

UNIVERSIDADE ABERTA



infantinho.xyz, um portal digital pedagógico

Igor Alexandre da Silva Godinho de Almeida

Trabalho de Projeto

Mestrado em Comunicação Educacional e Média Digitais

2018

UNIVERSIDADE ABERTA



infantinho.xyz, um portal digital pedagógico

Igor Alexandre da Silva Godinho de Almeida

Trabalho de projeto

Mestrado em Comunicação Educacional e Média Digitais

Trabalho de Projeto orientado pelo
Professor Doutor José Manuel Emiliano Bidarra de Almeida

2018

I'm gonna be like you, Dad.
You know I'm gonna be like you.
Harry Chapin

Resumo

Nos últimos anos, a disseminação de artefactos tecnológicos e a utilização de recursos online é uma constante nas salas de aula de Tecnologias da Informação e Comunicação(TIC). No entanto, os portais pedagógicos analisados parecem manter uma relação agnóstica com as correntes pedagógicas existentes. Neste projeto, é descrito o processo de desenvolvimento de um portal pedagógico agregador de recursos pedagógicos utilizados no contexto de pedagogias ativas. O portal é concebido para facilitar a adoção destas pedagogias ativas nas aulas da atividade de enriquecimento curricular TIC, no Pré-Escolar e Primeiro ciclo do Colégio Infante D. Henrique; convertendo para um modelo digital os instrumentos da gestão cooperada da sala de aula, como é o caso das listas de verificação dos conhecimentos e do diário de turma; facilitando o acesso aos documentos de apoio ao trabalho de aprendizagem por projetos cooperativos, de construção de cenários de aprendizagem e de planificação do trabalho autónomo; e mantendo um registo de todo o trabalho realizado, deste modo o portal pode torna-se num meio privilegiado para melhorar a comunicação, a difusão e a partilha dos conhecimentos construídos cooperativamente.

Palavras-chave: cenários de aprendizagem, artefactos tecnológicos, pedagogia ativa, portal pedagógico

Abstract

In recent years the spread of technological artefacts and the use of online resources is a constant in the Information and Communication Technologies classrooms (ICT), however, the learning portals analysed seems to maintains an agnostic relationship with existing pedagogical trends. In this project, we describe the process of developing a pedagogical portal that aggregates pedagogical resources, mostly used in the context of active pedagogies. The portal was designed to facilitate the adoption of these active pedagogies in the classes of curriculum enrichment activity ICT, in Reception to Year 4, in Colégio Infante D. Henrique. The goal is to convert to digital the instruments of cooperative classroom management, as is the case with the knowledge checklists and the class diary, and to facilitate the access to documents that help organize: learning by cooperative projects; building learning scenarios; and planning autonomous work. Keeping a record of all the work done, the portal can become a privileged means to a better communication, dissemination and share of knowledge built cooperatively.

Keywords: learning scenarios, technological artifacts, active learning, pedagogy portal

Agradecimentos

À Graciela pela paciência que teve e pelo apoio que me deu neste percurso sem o qual seria impossível tê-lo concluído.

À Lúcia e ao Lucas pelas brincadeiras, histórias, jogos, abraços e beijos que me sustentaram nesta caminhada.

Ao meu irmão que tantas vezes serviu de “pato de borracha” e me inspirou com a sua força de vontade e dedicação.

Aos meus pais, os verdadeiros culpados desta aventura pela Terra.

Aos meus familiares e amigos que estiveram sempre presentes.

Aos professores que me ensinaram a ser o profissional que hoje sou.

Índice Geral

Resumo.....	v
Abstract	vii
Agradecimentos	ix
Índice Geral	xi
Índice de Tabelas	xiii
Índice de Figuras	xv
1. Introdução	1
1.1. Objetivos	3
1.1.1. Objetivo geral.....	4
1.1.2. Objetivos específicos	4
1.2. Metodologia.....	4
2. Enquadramento e Contextualização	7
2.1. Atividade de Enriquecimento Curricular TIC na Madeira.....	8
2.2. Cenários de Aprendizagem – um Possível Caminho para o Futuro	9
2.3. Regulação das aprendizagens.....	10
3. Desenvolvimento do Portal infantinho.xyz	11
3.1. Análise de outros sítios com índole pedagógica	11
3.1.1. Google Classroom	11
3.1.2. ClassDojo.....	12

3.1.3. ClassFlow	13
3.2. Requisitos para o portal infantinho.xyz.....	14
3.3. Ferramentas e tecnologias	21
3.3.1. Linguagem de desenvolvimento do portal	21
3.3.2. Base de dados	23
3.3.3. Servidor de páginas de internet.....	23
3.3.4. Sistema operativo do hospedeiro	24
3.3.5. Serviço de hospedagem	24
3.3.6. Alojamento dos ficheiros estáticos	25
3.4. Processo de desenvolvimento.....	26
3.4.1. Configurações do Virtual Private Server	27
3.4.2. Blogue.....	36
3.4.3. Registo de Presenças.....	38
3.4.4. Listas de Verificação	43
3.4.5. Planos de Trabalho	45
3.4.6. Diário de Turma e Recursos	47
4. Considerações Finais	49
4.1. Conclusão	49
4.2. Aplicabilidade	51
Referências	53

Índice de Tabelas

Tabela 3.1. Lista com os Requisitos Funcionais — Organizados pelas Funcionalidades.....	15
Tabela 3.2. Lista com os Requisitos Não-Funcionais	20
Tabela 3.3. Exemplo de Configuração do Apache2	28
Tabela 3.4. Lista dos Comandos Utilizados para a Criação da Estrutura do <code>infantino.xyz</code> ..	29
Tabela 3.5. Exemplo de uma classe do Ficheiro <code>models.py</code>	31
Tabela 3.6. Exemplo de uma função do Ficheiro <code>views.py</code>	31
Tabela 3.7. Exemplo de um template	32
Tabela 3.8. Alterações ao Ficheiro <code>settings.py</code> para Configurar a Utilização do MySQL	32
Tabela 3.9. Alterações ao Ficheiro <code>settings.py</code> para Ativar a Utilização do Google Cloud Storage	33
Tabela 3.10. Alterações ao Ficheiro <code>settings.py</code> para Ativar a Utilização do Google OAuth2	34
Tabela 3.11. JavaScript para gosto e partilhar	37
Tabela 3.12. View para Carregar a Tabela e Criar um Utilizador para Cada Aluno	39
Tabela 3.13. Model da App que Guarda as Presenças na Base de Dados.....	41
Tabela 3.14. Models das Listas de Verificação	43
Tabela 3.15. Class Project.....	46
Tabela 3.16. Class Knowwhat.....	46

Índice de Figuras

Figura 3.1. Quadro com as funcionalidades apresentadas na página do Google Classroom.	12
Figura 3.2. Quadro com as funcionalidades apresentadas na página do ClassDojo	13
Figura 3.3. Quadro com as funcionalidades apresentadas na página do ClassFlow	14
Figura 3.4. Descrição esquemática do padrão MVC.	22
Figura 3.5. Descrição esquemática do padrão MTV.....	23
Figura 3.6. Rank dos motores de base de dados em 2015 de http://db-engines.com/en/ranking	23
Figura 3.7. Planos de máquinas virtuais oferecido pela digitalocean.com em dezembro de 2015	24
Figura 3.8. Planos de máquinas virtuais oferecido pela dreamhost.com em dezembro de 2015	25
Figura 3.9. Planos de máquinas virtuais oferecido pela linode.com em dezembro de 2015	25
Figura 3.10. Página inicial do infantinho.xyz	38
Figura 3.11. Django Admin App – Criar Disciplina	41
Figura 3.12. Django Admin App – Criar Professor	41
Figura 3.13. Tabelas Criadas pelo Django a Partir dos Models das Apps Student e Record	42
Figura 3.14. Marcar Presenças	43
Figura 3.15. Escolher o Módulo	44
Figura 3.16. Escolher o Nível de Dificuldade	45
Figura 3.17. Avaliar cada Competência Definida.....	45

1. Introdução

O século XXI coloca desafios fundamentais aos educadores, professores e pedagogos: a velocidade em que ocorrem as mudanças tecnológicas e sociais podem ser fonte de incertezas para todos os intervenientes no processo educativo e para o sistema educativo em si mesmo. Neste sentido, urge “criar «espaço-tempo» de formação” nos quais as crianças têm de “aprender a servir-se dos saberes adquiridos para enfrentar situações novas e singulares” (Perrenoud, 2002, pp. 142–143).

Um dos principais desafios prende-se com a relação entre o acesso ao conhecimento, que estava centralizado nas instituições escolares e na figura do professor, “o mestre”, e a partilha de conhecimento, que é construído diariamente na internet através de diferentes médias. “In such a world, many will only dabble, some will dig deeper, and still others will master the skills that are most valued within the community. The community itself, however, provides strong incentives for creative expression and active participation.” (Jenkins, 2006)

As mudanças tecnológicas e sociais, decorrentes, sobretudo, do relacionamento dos indivíduos com a internet, colocam ainda mais pressão nos profissionais da educação no sentido de colmatarem algumas lacunas. Jenkins (2006), refere a necessidade de uma intervenção pedagógica em três preocupações:

The Participation Gap — the unequal access to the opportunities, experiences, skills, and knowledge that will prepare youth for full participation in the world of tomorrow.

The Transparency Problem — The challenges young people face in learning to see clearly the ways that media shape perceptions of the world.

The Ethics Challenge — The breakdown of traditional forms of professional training and socialization that might prepare young people for their increasingly public roles as media makers and community participants. (Jenkins, 2006, p. 3)

E conclui que é preciso mudar o foco da discussão pedagógica sobre o acesso ao mundo digital das questões tecnológicas para as questão de “*opportunities to participate and to*

develop the cultural competencies and social skills needed for full involvement”(Jenkins, 2006, p. 4).

Neste contexto, o professor tem de assumir uma atitude de abertura perante o conhecimento, procurando sempre construir com os seus alunos o sentido do saber, organizando e criando situações de aprendizagens, avaliando-os formativamente e implicando-os em todos os passos deste processo, respeitando e valorizando as suas diferenças individuais, regulando todo o percurso de formação e refletindo sobre a sua própria prática pedagógica.

O Ensino continua a ter um papel fundamental no sucesso dos indivíduos, dos países e das corporações e *“growing evidence demonstrates that—among all educational resources—teachers’ abilities are especially crucial contributors to students’ learning.”(Darling-hammond, 2006, p. 1)* Assim, atendendo, por um lado, à necessidade de construir um conhecimento fundamentado, e, por outro, ao novo paradigma de sociedade participativa, *“importa criar condições de equilíbrio entre o conhecimento, a compreensão, a criatividade e o sentido crítico.”(Ministério da Educação, 2017, p. 6)*

No sentido de orientar o trabalho dos professores de modo a que estes sejam capazes de *“preparar para o imprevisto, o novo, a complexidade e, sobretudo, desenvolver em cada indivíduo a vontade, a capacidade e o conhecimento que lhe permitira aprender ao longo da vida”(Ministério da Educação, 2017)*, o Ministério da Educação (MEC), no documento para consulta pública, *Perfil dos Alunos para o Século XXI*, refere, logo nos princípios que os orientou na construção do referido documento, a necessidade de:

- valorizar a nossa herança humanista — que vê cada aluno como uma pessoa única e insubstituível, a qual deve ter tempo e espaço para exercer a sua cidadania e, no entanto, promover o grupo como o lugar ideal de desenvolvimento;
- ensinar a aprender — ao construir o conhecimento com os alunos, o professor deve demonstrar que o conhecimento é uma construção constante e que é feito ao longo da vida;
- valorizar a diferença — todos temos de perceber que *“o que vos sobra a vós supera a indigência dos outros, para que um dia o supérfluo deles compense*

a vossa indignação” (2 Cor. 8:14 Bíblia dos Capuchinhos) e que, neste sentido, todos têm algo de valioso para a nossa aula;

- atentar ao ambiente — as preocupações com as questões da sustentabilidade do nosso planeta devem ser uma constante;
- ser coerente e flexível — é na relação com os colegas que a comunidade, turma, se desenvolve, o professor tem, portanto, de pôr a tónica *“on communities as learners, on transformation and creation of culture, on horizontal movement and hybridization and on the formation of theoretical concepts”*(Engeström, 2016, p. 36);
- agir com adaptabilidade e ousadia — procurando, por um lado, incluir novos paradigmas na prática pedagógica e, por outro, aceitando a inclusão de novos artefactos pedagógicos.

O professor, para ser capaz de representar, de forma cabal, todos estes papéis que lhe são pedidos, precisa, necessariamente, de ferramentas que o ajude na organização prática destes princípios pedagógicos. Neste sentido, com o intuito de fornecer ferramentas adequadas ao desenvolvimento da prática docente, pretende-se criar um portal agregador de: artefactos tecnológicos produzidos pelo professor e pelos alunos; ferramentas que facilitem o processo de regulação das aprendizagens; guiões para a criação de projetos ou cenários de aprendizagem; e de divulgação dos acontecimentos da escola.

1.1. Objetivos

Uma vez que a regulação das aprendizagens tem um papel fundamental no desenvolvimento integral dos alunos, este projeto tem por objetivo criar uma **plataforma digital** que reúna e operacionalize neste formato os diferentes instrumentos de regulação do grupo e das aprendizagens, de modo a que os alunos sejam capazes de:

- Regular as suas aprendizagens com o apoio e feedback do professor e dos seus pares;
- Realizar atividades adequadas aos seus níveis de desenvolvimento;
- Dar sentido ao seu trabalho com as novas tecnologias;
- Descobrir novos conhecimentos através da utilização das TIC;
- Pensar criticamente para a resolução de problemas reais.

1.1.1. Objetivo geral

Criar uma plataforma digital agregadora de artefactos pedagógicos potenciadores e facilitadores das aprendizagens STEM (*science, technology, engineering and mathematics*); de ferramentas para a regulação do processo de ensino/aprendizagem; e de divulgação dos trabalhos e atividades realizadas pelos alunos do Pré-Escolar e do Primeiro Ciclo do Colégio Infante D. Henrique.

1.1.2. Objetivos específicos

Para que se possa operacionalizar o objetivo geral enunciado, foram identificados alguns objetivos para plataforma digital:

- A divulgação dos trabalhos;
- A organização e a categorização dos artefactos pedagógicos produzidos;
- A criação, gestão e regulação de projetos de grupo;
- A criação, gestão e regulação dos trabalhos individuais;
- A regulação das presenças, das aprendizagens e dos comportamentos.

1.2. Metodologia

Uma metodologia de trabalho bem definida pode contribuir decisivamente para um bom desenrolar do projeto. Por um lado, em termos pedagógicos e do desenvolvimento profissional procurar-se-á seguir um modelo espiral de Ação—Reflexão—Ação, no qual, partindo da realidade fazem-se opções, buscando a transformação da realidade. Uma vez realizada a ação é indispensável refletir, a fim de aferir se os objetivos foram atingidos e se os meios eram, ou não, adequados. Este modelo de não deixa cristalizar nos resultados obtidos, mas exige um aprofundamento e uma entrega sempre maiores. Por outro lado, em termos do desenvolvimento do software, também é necessário haver um modelo de trabalho bem definido.

De um modo diluto, procurou-se seguir uma metodologia *agile* com uma tónica no trabalho de “*a single programmer working alone [...] in that case, the priorities and needs change, communication takes another meaning and working mechanisms are not always comparable to that of a team.*” (Bernabé, Navia, & García-Peñalvo, 2015) Assim, escolheu-se seguir a metodologia *Freelance as a Team* (FaaT).

Nesta metodologia estão definidas *estratégias, fluidez de trabalho e práticas auxiliares*. As práticas estrategicamente defendidas por esta metodologia são:

- Promover a simplicidade — procurar sempre a solução mais simples, aquela que produzirá o mínimo produto viável, sobre o qual deverão ser acrescentadas apenas as funcionalidades necessárias;
- Aceitar a Mudança — a mudança deve ser planeada e pensada à priori, o código não deve ser escrito de uma maneira que impossibilite a sua alteração. Pelo contrário, as alterações devem ser fáceis de implementar e para tal a utilização de ferramentas de controle de versões é essencial;
- Tomar decisões — embora se deva promover a mudança e aceitá-la, o processo de tornar o código flexível não deve ocupar o tempo de programador, não perdendo tempo com problemas que ainda não surgiram.

No que diz respeito à fluidez do trabalho, a metodologia FaaT, define algumas práticas que podem contribuir positivamente no desenvolvimento ágil de software:

- Definir as *funcionalidades*, principalmente, na perspetiva do *utilizador* e que acrescentem valor ao utilizador;
- Fazer uma *estimativa* do *tempo* necessário para o desenvolvimento de cada funcionalidade, quando trabalhando sozinho, o desenvolvedor deve “*measure the actual effort spent on each task and after each iteration compare estimations against actual effort and draw conclusions for future estimates*”(Bernabé et al., 2015);
- Criar *prioridades de desenvolvimento*, agrupando as funcionalidades tematicamente e completando cada funcionalidade até ao fim do prazo estipulado, “*agile projects must be flexible in scope and rigid in timing*”(Bernabé et al., 2015);
- Manter um *registo* das funcionalidades, erros encontrados, manutenção do sistema realizada e conhecimentos que precisam de ser aprofundados;
- *Testar* todas as interações possíveis, de preferências de uma maneira automática e se possível antes do desenvolvimento do produto, ou seja, testar o conceito;

- Utilizar um *sistema de controle de versões*, que permita a criação automática de backups, a manutenção de diferentes versões, comparar o código, voltar a uma versão estável em caso de um erro grave, e que permita rever a história de criação de código;
- Avaliar, reavaliar, melhorar;
- Medir o tempo investido em cada funcionalidade, a percentagem do projeto que já está pronto e percentagem das funcionalidades com testes automáticos.

Por fim, na metodologia de desenvolvimento FaaT algumas práticas auxiliares do trabalho podem ser adotadas. A *reutilização do código*, aproveitando classes e funções e organizando-as de acordo com as suas funcionalidades. Criar apenas o *mínimo* necessário de *documentação*. Desenvolver *protótipos parciais* para explorar novas soluções. Adquirir um *rubber duck*, como um

“highly effective and very low cost debugging technique. It involves following four steps:

1. *Obtain by any mean (preferably legal) a rubber duck.*
2. *Place rubber duck on desk and inform it you are just going to go over some code with it, if that’s all right.*
3. *Explain to the duck what your code is supposed to do, and then go into detail and explain your code line by line.*
4. *At some point you will tell the duck what you are doing next and then realise that is not in fact what you are actually doing. The duck will sit there serenely, happy in the knowledge that it has helped you on your way.”*(Bernabé et al., 2015)

Por fim, tentar automatizar o máximo possível qualquer prática explicada.

No desenvolvimento do portal *infantinho.xyz* será utilizada a metodologia FaaT, embora nem todos os aspetos sejam adotados pois falta ao desenvolvedor a experiência e a prática para o fazer.

2. Enquadramento e Contextualização

“Os alunos aprendem melhor quando o professor toma em consideração as características próprias de cada um, visto que cada indivíduo possui pontos fortes, interesses, necessidades e estilos de aprendizagem diferentes.” (Grave-Resendes & Soares, 2002, p. 14) No entanto, para haver uma verdadeira diferenciação pedagógica o professor precisa de estar apetrechado com ferramentas que lhe permitam dar um efetivo apoio aos alunos com maior dificuldade. No modelo do Movimento da Escola Moderna (MEM), o professor parte das necessidades e interesses dos alunos e partilha com eles a gestão dos tempos, dos recursos e dos conteúdos escolares, envolvendo e responsabilizando-os pela sua aprendizagem.

Quem já observou uma sala de aulas a funcionar com o modelo pedagógico do MEM sabe que para que o modelo seja eficiente existe uma panóplia de artefactos de apoio à gestão partilhada, os quais, são de fácil acesso a todos. No entanto, num contexto de uma atividade de enriquecimento curricular, como é o caso das TIC na Região Autónoma da Madeira, a sala de aulas é partilhada por todas as turmas da escola. No caso do Colégio Infante D. Henrique, no ano letivo de 2016/2017 são sete turmas que partilham o laboratório dos computadores, tornando-se praticamente impossível haver materiais, específicos de cada turma, disponíveis. Neste sentido, a conversão destas ferramentas para um modelo digital é necessária e tem sido feita com o recurso a diferentes sítios da internet, sobretudo, através das ferramentas de trabalho cooperativo da Google APPS: Google Drive, Documentos do Google, Folhas de Cálculo do Google e Apresentações do Google.

Com a utilização do *infantino.xyz* os alunos poderão escolher, com o auxílio do professor, quais os trabalhos que realizarão e que curiosidade ou aspeto do currículo vão trabalhar. “A curiosidade no sentido epistemológico é um adequar-se entre mente e circunstâncias; neste adequar-se o que se pretende é conhecer a razão de ser dos” (Freire & Freire, 2001, p. 187) assuntos escolhidos pelos alunos e que serão tornados comuns através das apresentações aos colegas dos trabalhos realizados. É de notar que estes trabalhos podem assumir diferentes formas, mas que, de uma maneira geral, implicarão a construção de um artefacto tecnológico que passará a estar agregado no portal.

A publicação destes artefactos no portal permitirá aos colegas, professores e pais tecerem comentários que podem ajudar os alunos a se desenvolverem e a atribuírem um maior sentido aos seus trabalhos. Nestes comentários, o professor deve assumir “a postura de um amigo crítico, sem concessões, que procura não ser negativo, mas que também não fecha os olhos para as falhas de raciocínio ou fraqueza das observações.” (Perrenoud, 2002, p. 51)

O portal será assim um ponto de partida e um agregador de conhecimento partilhado e construído quer pelos alunos, quer pelo professor, tornando-se assim numa ferramenta central da gestão da sala de aula; bem como num elo de ligação entre as famílias e a escola. Tomando conhecimento do trabalho realizado na escola pelos seus filhos, os pais podem intervir, fornecendo-lhes o devido e atento acompanhamento.

2.1. Atividade de Enriquecimento Curricular TIC na Madeira

A Atividade de Complemento Curricular de Tecnologias da Informação e Comunicação (TIC) é lecionada no 1.º Ciclo do Ensino Básico na Madeira como uma atividade de enriquecimento curricular, no âmbito da escola a tempo inteiro. Neste contexto, o professor das TIC é chamado a preparar os seus alunos para a certificação de competências básicas, a qual é obtida através da realização de um exame final de ciclo, e que, segundo Documento de Orientações Metodológicas [DOM], constitui uma garantia e é “uma condição imprescindível para atingir maiores níveis de literacia digital e tecnológica na sociedade”(2010, p. 3).

O mesmo documento, no entanto refere que “as crianças se envolvem, aprendem e desenvolvem uma disposição para aprender recorrendo às TIC”. Neste sentido, parece importante haver instrumentos de regulação que permitam ao professor e aos alunos regular as aprendizagens específicas das TIC, mas com conteúdos de outras áreas do currículo.

O DOM parece sugerir a compartimentação dos conhecimentos necessários às TIC, em diferentes módulos, sendo que no 4.º ano são propostos os seguintes: Processador de Texto; Tratamento de imagens; Internet; Segurança na Internet; Correio Eletrónico; Apresentações Multimédia; Folha de Cálculo; Criação/manutenção de um blogue; Criação/edição de páginas Web online; Programa para Publicação; E-portfolio e Edição de

Vídeo. Embora essa divisão não seja de todo obrigatória, ela pode acabar por orientar a planificação do professor para um trabalho compartimentado que tem pouca relação com as necessidades reais dos alunos. Neste sentido, procuraremos no portal e na prática pedagógica desenvolvida com os nossos alunos, o desenvolvimento de atividades complexas que impliquem não apenas a utilização das diferentes ferramentas tecnológicas propostas no DOM bem como a utilização de outras que vão sendo incluídas a partir das necessidades dos alunos.

Partindo destes trabalhos complexos, pretender-se-á promover uma diferenciação pedagógica positiva que permita aos alunos com maior destreza avançarem nas suas descobertas e, por outro lado, que possibilite aos alunos com mais dificuldade uma aprendizagem ao seu ritmo.

2.2. Cenários de Aprendizagem – um Possível Caminho para o Futuro

Os cenários de aprendizagem são entendidos, no contexto do *Innovative Technologies for an Engaging Classroom* (iTEC) por Cranmer & Perrotta (2011, p. 4), como pequenas narrativas que descrevem contextos de aprendizagem, nos quais se enquadra um modelo de aprendizagem ativa e onde é explícito o que acontece nos cenários, onde este é desenvolvido, quem participa, como participa e quais os recursos necessários para a sua aplicação (What are scenarios in iTEC?, para. 1).

A criação destes cenários de aprendizagem pressupõe um trabalho cooperativo entre os seus diferentes intervenientes que lhes possibilita uma reflexão mais cuidada e verdadeira sobre as suas práticas pedagógicas, o seu desenvolvimento escolar e o seu papel na comunidade educativa.

Do ponto de vista do aluno, a participação nestes cenários permite uma intervenção ativa em todo o seu processo de criação e de desenvolvimento, o que, em última análise, poderá ser um meio para desenvolver “a vontade, a capacidade e o conhecimento que lhes permitirá aprender ao longo da vida” (Ministério da Educação, 2017, p. 8). Neste sentido, as crianças são agentes ativos da sua aprendizagem e participam construtivamente no desenvolvimento dos seus conhecimentos, no seu processo de avaliação, na avaliação dos seus pares e professores, na escolha dos saberes e dos artefactos produzidos e na necessária diferenciação pedagógica.

2.3. Regulação das aprendizagens

A avaliação formativa é o principal modelo de avaliação necessário à regulação das aprendizagens. A lei de bases do sistema educativo português considera-a como “a principal modalidade de avaliação e permite obter informação privilegiada e sistemática nos diversos domínios curriculares, devendo fundamentar o apoio às aprendizagens, nomeadamente à autorregulação dos percursos dos alunos em articulação com dispositivos de informação dirigidos aos encarregados de educação.”(Diário da República, 1.ª série—N.º 65— 4 de abril de 2016) No mesmo documento, a avaliação formativa é definida como aquela que:

Assume carácter contínuo e sistemático, devendo recorrer a uma variedade de instrumentos de recolha de informação adequados à diversidade das aprendizagens e às circunstâncias em que ocorrem, permitindo aos professores, aos alunos, aos encarregados de educação e a outras pessoas ou entidades legalmente autorizadas obter informação sobre o desenvolvimento do ensino e da aprendizagem, com vista ao ajustamento de processos e estratégias.(Diário da República, 1.ª série—N.º 65— 4 de abril de 2016)

Por sua vez, o projeto curricular de escola do Colégio Infante D. Henrique(2015), define que a avaliação das atividades de enriquecimento curricular “implica uma apreciação qualitativa em registo próprio” e que “tem-se em conta a assiduidade, interesse e participação”.

3. Desenvolvimento do Portal *infantino.xyz*

O processo de desenvolvimento tem avanços e recuos, escolhas e imposições que o fazem ser dinâmico e fluído e, portanto, capaz de dar resposta às necessidades que surgem nas aulas e na utilização do portal.

A criação deste portal só faz sentido se trazer benefícios práticos à solução alternativa, a utilização dos materiais em formato de papel, o facto de os dados estarem em formato digital, permite um tratamento de dados mais simplificado e uma quantidade virtualmente infinita de recursos, i.e., não limitada ao número de fotocópias disponível para cada professor.

3.1. Análise de outros sítios com índole pedagógica

Nos últimos anos têm surgido diferentes plataformas de aprendizagem online ou de apoio à aprendizagem na sala de aula. No entanto, as plataformas analisadas revelaram-se, por um lado, demasiado complexas para serem aplicadas nas aulas de TIC do 1.º ciclo, quer pela barreira linguística, quer pela curva de aprendizagem para o utilizador, ou ainda, pelos requisitos mínimos de hardware. Por outro lado, revelaram-se insuficientes nas características disponíveis, principalmente pela ausência de meios de divulgação dos trabalhos realizados à comunidade educativa.

Uma solução possível poderia passar pela utilização de vários serviços de forma a completar as lacunas encontradas nestas plataformas. No entanto, parece ser preferível a possibilidade de agregar a utilização de alguns destes serviços a partir de um único portal com a necessidade de se identificar apenas uma vez e que possa ser desenvolvido e alargado conforme as necessidades do grupo de trabalho.

3.1.1. Google Classroom

O Google Classroom (<http://classroom.Google.com>) é um serviço que permite aos professores estarem conectados com os seus alunos.

A integração com as outras ferramentas da Google são uma grande mais-valia deste serviço, permitindo a criação de documentos modelos e uma fácil gestão dos trabalhos atribuídos aos alunos.

Usuário	O que você pode fazer com o Google Sala de aula
Professores	<ul style="list-style-type: none">• Criar e gerenciar turmas, tarefas e notas.• Dar feedback direto e em tempo real e atribuir notas.
Alunos	<ul style="list-style-type: none">• Acompanhar os materiais e as tarefas da turma.• Compartilhar recursos e interagir no mural da turma ou por e-mail.• Enviar tarefas.• Receber notas e feedback.
Responsáveis	<p>Receber um resumo por e-mail sobre os trabalhos do aluno. Esse resumo inclui informações sobre trabalhos pendentes, próximas tarefas e atividades da turma.</p> <p>Observação: os responsáveis não podem fazer login diretamente no Google Sala de aula. Eles precisam receber os resumos por e-mail por outra conta.</p>
Administradores	<ul style="list-style-type: none">• Criar, ver ou excluir qualquer turma no domínio.• Adicionar ou remover alunos e professores das turmas.• Ver os trabalhos de todas as turmas do domínio.

Figura 3.1. Quadro com as funcionalidades apresentadas na página do Google Classroom.

Alguns pontos negativos identificados são a falta de um espaço para a divulgação dos trabalhos realizados; o pressuposto de que todos os alunos farão o mesmo trabalho; e a ausência de ferramentas para criar e gerir grupos de trabalho.

3.1.2. ClassDojo

ClassDojo (<http://www.classdojo.com>) é uma plataforma de gestão da sala de aula com ferramentas para atribuir pontos positivos para comportamentos, atitudes e trabalhos realizados, e pontos negativos para maus comportamentos, atitudes indesejáveis ou trabalhos não realizados.

A facilidade de utilização, o feedback imediato aos alunos e encarregados de educação e a possibilidade de aceder ao serviço através de diferentes plataformas foram os aspetos positivos identificados neste serviço.

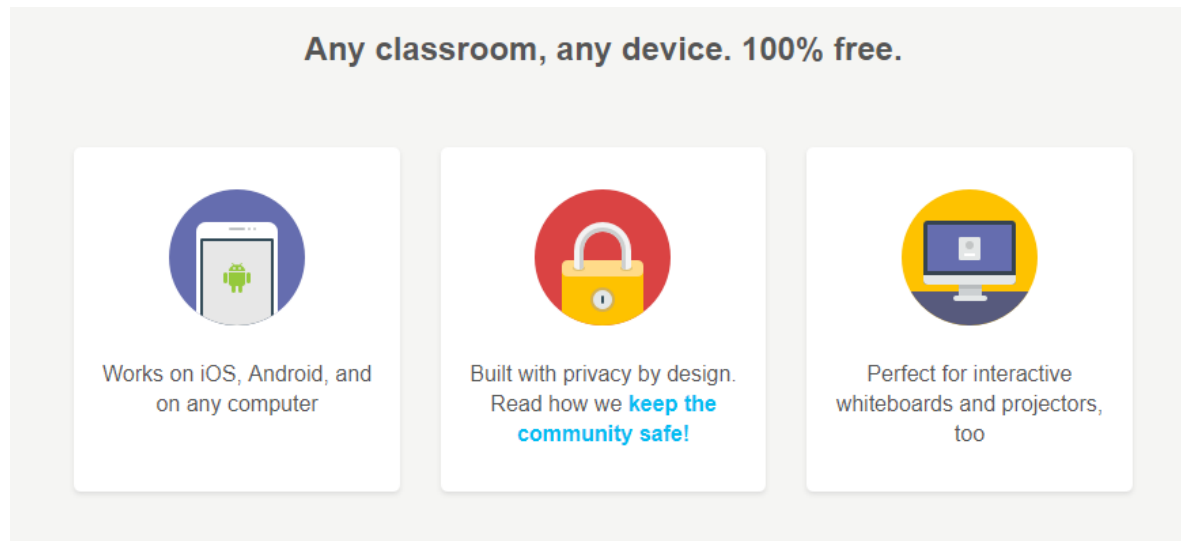


Figura 3.2. Quadro com as funcionalidades apresentadas na página do ClassDojo

Alguns pontos negativos identificados são a falta de um espaço para a divulgação dos trabalhos realizados, a ausência de ferramentas para criar e gerir grupos de trabalho e a falta de conteúdos em português.

3.1.3. ClassFlow

Classflow (<http://classflow.com>) é uma plataforma educativa que foi criada pela Promethean, uma das principais produtoras de quadros interativos, para o desenvolvimento de aulas interativas a serem utilizadas em quadros interativos.

A possibilidade de ser utilizado com qualquer dispositivo interativo, o feedback imediato, as ferramentas de perguntas interativas, a possibilidade de partilhar o ecrã de qualquer computador para o quadro interativo, a gestão dos trabalhos de casa e a aplicação de testes e questionários para grupos de alunos foram as potencialidades identificadas.

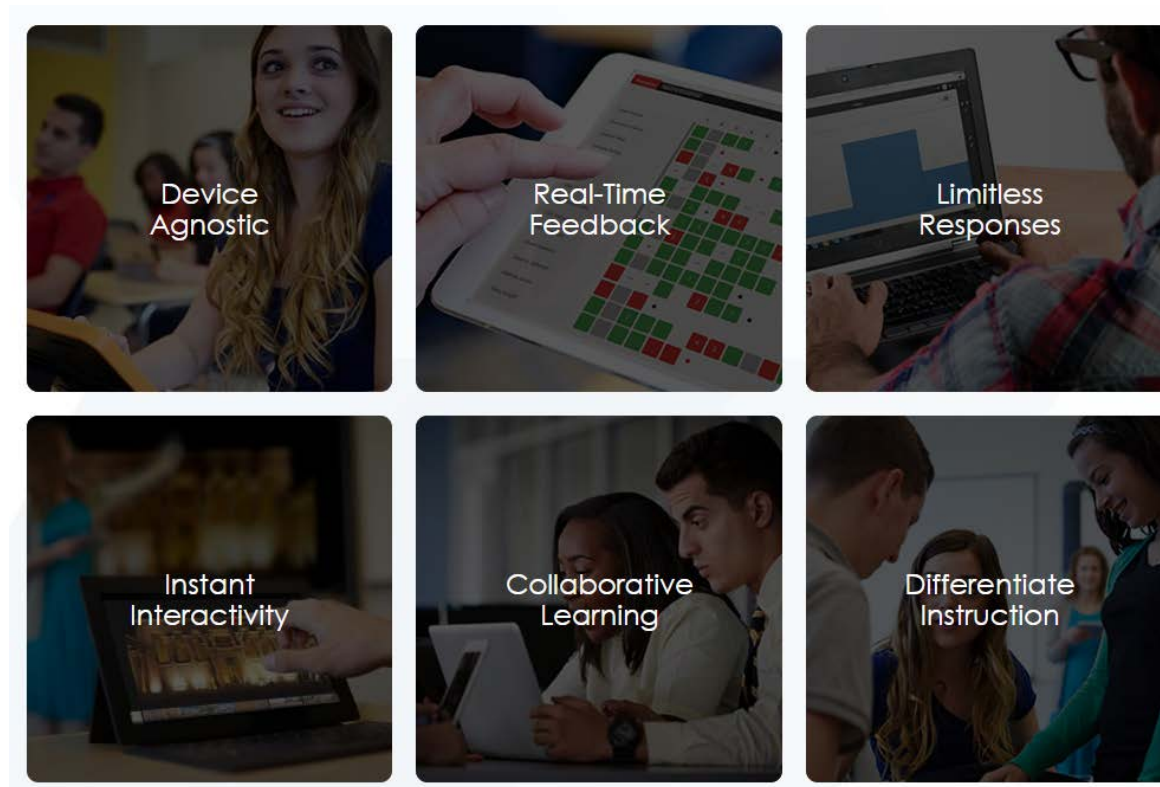


Figura 3.3. Quadro com as funcionalidades apresentadas na página do ClassFlow

A necessidade de utilizar quadros interativos, que não existem de momento na nossa escola, a falta de um espaço para a divulgação dos trabalhos realizados, a ausência de ferramentas para criar e gerir grupos de trabalho e a falta de conteúdos em português foram os pontos negativos identificados nesta plataforma.

3.2. Requisitos para o portal *infantino.xyz*

Tendo em conta os objetivos do portal *infantino.xyz* e os seus utilizadores, foram definidos os seus requisitos de forma a que pudessem ser utilizados como linhas orientadoras no desenvolvimento.

Estes requisitos podem e devem ser adaptados ao longo do processo de desenvolvimento para dar resposta às necessidades específicas dos utilizadores do portal. Os requisitos podem ser organizados em funcionais, descritos na Tabela 3.1., e não funcionais, descritos na Tabela 3.2..

Tabela 3.1. Lista com os Requisitos Funcionais — Organizados pelas Funcionalidades

Nome	Requisito
Navegação	O utilizador deve ser capaz de navegar pelas hiperligações no <i>infantinho.xyz</i> .
BASE DE DADOS	
Identidade	O utilizador deve ser capaz de identificar-se no <i>infantinho.xyz</i> .
Identidade:Verificar	O <i>infantinho.xyz</i> deve ser capaz de verificar o nível de acesso do utilizador, identificando-os como utilizador, convidado, aluno e professor.
Identidade:DarAcesso	O <i>infantinho.xyz</i> deve dar acesso aos conteúdos de acordo com o nível de acesso atribuído, sendo que: <ul style="list-style-type: none">- o professor tem acesso total;- o aluno tem acesso a uma parte do conteúdo disponível ao professor;- o convidado tem acesso a uma parte do conteúdo disponível ao aluno;- o utilizador tem acesso a uma parte do conteúdo disponível ao convidado.
Relações	O <i>infantinho.xyz</i> deve ser capaz de relacionar os utilizadores com seus conteúdos.
Turmas:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar turmas.
Disciplinas:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar disciplinas.
Alunos:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar alunos.
Relações:Turmas, Disciplinas e Alunos	O <i>infantinho.xyz</i> deve ser capaz de relacionar turmas, disciplinas e alunos.

BLOGUE

Blogue:VP	O utilizador deve ser capaz de visualizar e partilhar, nas redes sociais, as mensagens do blogue.
Blogue:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar mensagens no blogue.
Blogue:Rascunhos:VEIA	O aluno deve ser capaz de visualizar, excluir, inserir e alterar os seus rascunhos de mensagem no blogue.
Blogue:Rascunhos:VEIAA	O professor deve ser capaz de visualizar, excluir, alterar e aprovar rascunhos para mensagens no blogue.
Blogue:DB:Relacionar	O infantinho.xyz deve ser capaz de relacionar mensagens e comentários com os seus autores.
Blogue:Mensagem:Edição	O infantinho.xyz deve ser capaz de apresentar ferramentas de edição de texto.
Blogue:Mensagem:Média	O infantinho.xyz deve ser capaz de apresentar ferramentas de inclusão de diferentes tipos de mídias digitais, i.e., imagem, vídeo e conteúdo incorporado.

REGISTOS

Presenças:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar presenças.
Presenças:Relatórios	O infantinho.xyz deve ser capaz de produzir relatórios de presenças de um dado intervalo temporal.
Presenças:Relatórios:XMLX	O infantinho.xyz deve ser capaz de produzir em formato Excel, no modelo utilizado pelo Colégio Infante D. Henrique, relatórios com as presenças diárias, mensais e anuais de uma determinada disciplina.
Presenças:Relatórios:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar relatórios de presenças.
Regulação:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar modelos de listas de verificações com os descritores de competências espectáveis.

Regulação:VIA	O aluno deve ser capaz de visualizar, inserir e alterar os valores das suas listas de verificação.
Regulação:Relações	O infantinho.xyz deve ser capaz de relacionar descritores de competência, turmas e alunos.
Regulação:Relatórios	O infantinho.xyz deve ser capaz de produzir relatórios com os valores das listas de verificação por turmas, alunos ou descritores.
Regulação:Relatórios:V	O aluno deve ser capaz de visualizar os seus relatórios das listas de verificação.
Regulação:Relatórios:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar os relatórios das listas de verificação.
Regulação:Ajuda	O infantinho.xyz deve ser capaz de apresentar uma lista de alunos/professores disponíveis para ajudar num determinado descritor.
Regulação:Ajuda:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar os alunos/professores disponíveis para ajudar num determinado descritor.
Regulação:Ajuda:VI	O aluno deve ser capaz de visualizar e se inserir na lista de alunos disponíveis para ajudar num determinado descritor.
Planos:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar modelos de plano de trabalho individual e de projeto.
Planos:Individuais:VIA	O aluno deve ser capaz de visualizar, inserir e alterar os seus planos individuais de trabalho.
Planos:Projetos:VIA	O aluno deve ser capaz de visualizar, inserir e alterar os seus planos de trabalho de projeto.
Planos:Relações	O infantinho.xyz deve ser capaz de relacionar alunos, planos e produtos.

Planos:Relatórios	O infantinho.xyz deve ser capaz de produzir relatórios informativos com os planos individuais e os planos de projeto.
Planos:Relatórios:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar os relatórios dos planos individuais e dos planos de projeto.
Planos:Relatórios:V	O aluno deve ser capaz de visualizar os seus planos individuais e de projeto.
Diários:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar modelos de diários de turma.
Diários:VI	O aluno deve ser capaz de visualizar e inserir valores nos diários de turma.
Diários:Relações	O infantinho.xyz deve ser capaz de relacionar alunos e diários.
Diários:Relatórios	O infantinho.xyz deve ser capaz de produzir relatórios com informativos com as entradas nos diários de turma.
Diários:Relatórios:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar os relatórios dos diários de turma.
Diários:Relatórios:V	O aluno deve ser capaz de visualizar os relatórios dos diários de turma.
PROJETOS	
Projetos:Produtos:VEIA	O aluno deve ser capaz de visualizar, excluir, inserir e alterar os seus produtos dos projetos.
Projetos:Produtos: Avaliação:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar modelos para a avaliação dos produtos dos projetos.
Projetos:Produtos: Avaliação:VIA	O aluno deve ser capaz de visualizar, inserir e alterar os valores da avaliação dos produtos dos projetos.
Projetos:Apresentações: VEIA	O aluno deve ser capaz de visualizar, excluir, inserir e alterar os recursos para a apresentações dos projetos.

Projetos:Apresentações: HAvaliação:VEIA	O aluno deve ser capaz de visualizar, excluir, inserir e alterar os seus modelos para a hétero avaliação dos conhecimentos sobre a apresentação do seu projeto.
Projetos:Apresentações: HAvaliação:VI	O aluno deve ser capaz de visualizar e inserir valores na sua avaliação dos conhecimentos sobre a apresentação dos projetos dos colegas.
Projetos:Apresentações: AAvaliação:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar modelos de autoavaliação de apresentações de projetos.
Projetos:Apresentações: AAvaliação:VI	O aluno deve ser capaz de visualizar e inserir valores na sua autoavaliação do seu projeto.
Projetos:Relatórios	O infantinho.xyz deve ser capaz de produzir relatórios informativos com base nas avaliações dos projetos.
Projetos:Relatórios:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar os relatórios dos projetos.
Projetos:Relatórios:V	O Aluno deve ser capaz de visualizar os relatórios dos projetos.
Projetos:Relações	O infantinho.xyz deve ser capaz de relacionar alunos, grupo de trabalho de projeto, produtos, hétero avaliações, autoavaliações e avaliações dos projetos.
Projetos:Corrente	O infantinho.xyz deve ser capaz de produzir um <i>feed</i> de notícias com todos os títulos e os produtos dos projetos.
RECURSOS	
Recursos:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar as categorias dos artefactos digitais disponíveis.
Recursos:Artefactos:VEIA	O professor deve ser capaz de visualizar, excluir, inserir e alterar artefactos às categorias.
Recursos:Artefactos:VI	O aluno deve ser capaz de visualizar e inserir artefactos às categorias.
Recursos:Artefactos:V	O utilizador deve ser capaz de visualizar artefactos.

Recursos:Artefactos:Mostrar	O infantinho.xyz deve ser capaz de mostrar, organizados por categorias, os artefactos disponíveis.
Recursos:Artefactos:Relações	O infantinho.xyz deve ser capaz de relacionar artefactos, descritores de competências, alunos, hiperligações locais e remotas.

Tendo em conta os requisitos funcionais, os recursos disponíveis e os utilizadores espectáveis foram definidos os requisitos não funcionais que poderão dar uma melhor resposta às necessidades do projeto.

Tabela 3.2. Lista com os Requisitos Não-Funcionais

Nome	Requisito
Custo¹	O sistema de alojamento e desenvolvimento do infantinho.xyz deve ter um custo aproximado de zero.
Performance	O sistema de alojamento e desenvolvimento do infantinho.xyz deve estar disponível para ser acedido sempre, com um tempo de resposta curto e um consumo de largura de banda reduzido.
Documentação	O sistema de alojamento e desenvolvimento do infantinho.xyz deve estar bem documentado, com exemplos de utilização prática e guias de implementação.
Cópias de Segurança	O sistema de alojamento e desenvolvimento do infantinho.xyz deve ter um processo de cópias de segurança automático.
Segurança	O sistema de alojamento e desenvolvimento do infantinho.xyz deve ter um sistema robusto de segurança da informação dos utilizadores e do código do portal.
Disponibilidade	O portal infantinho.xyz deve estar disponível para o maior número de dispositivos possíveis.

¹ Nenhuma verba foi alocada pelo Colégio Infante D. Henrique para este projeto.

Reusabilidade	O portal <i>infantinho.xyz</i> deve ser fácil de replicar em qualquer sistema operativo, na nuvem ou localmente.
Usabilidade	O portal <i>infantinho.xyz</i> deve ser fácil de utilizar por crianças do primeiro ciclo e pré-escolar, não permitindo erros catastróficos e sendo capaz de corrigir dados inválidos sem a interferência do utilizador.

3.3. Ferramentas e tecnologias

Tendo como pano de fundo os requisitos não funcionais enunciados importa escolher: um serviço de hospedagem; o sistema operativo do hospedeiro; linguagem de desenvolvimento para o portal; base de dados; servidor de páginas da internet; e o alojamento dos ficheiros estáticos.

3.3.1. Linguagem de desenvolvimento do portal

As linguagens de programação *“can be used in web development to make websites more dynamic and to handle advanced logic beyond simply returning static HTML”*(Swain et al., 2015, p. 112). Tendo em conta os requisitos não funcionais que foram definidos, importa escolher uma linguagem de programação de fácil aprendizagem, bem documentada com guias e exemplos de implementação e open source.

Swain et al.(2015), refere que as linguagens mais comuns no desenvolvimento de web apps identificadas foram JAVA, PHP e Python. Por outro lado, Meyerovich & Rabkin(2013) identificaram PHP e Python linguagens de rápida aprendizagem, cerca de 3 meses, enquanto que JAVA como uma linguagem de aprendizagem mediana, cerca de 6 meses. Ainda no mesmo estudo, Meyerovich & Rabkin referem que *“PHP is notorious for its ad-hoc design while Python well-regarded for its simplicity”*.

Com a informação recolhida foi possível escolher Python como a linguagem de desenvolvimento para o portal *infantinho*. Decidiu-se também pela utilização da framework Django, uma vez que esta framework pressupõe, e implementa automaticamente, uma estrutura de desenvolvimento semelhante ao padrão de arquitetura de software *model-view-controller* (MVC), no qual o desenvolvimento das aplicações é dividido em três tipos de componentes: *model, views e controllers*.

Hansson (2005), apresenta um esquema claro do papel de cada um dos componentes do padrão MVC, Figura 3.4., por sua vez, a framework Django “*follows the MVC pattern closely, however it does use its own logic in the implementation. Because the «C» is handled by the framework itself and most of the excitement in Django happens in models, templates and views, Django is often referred to as an MTV framework*”(George, 2016).

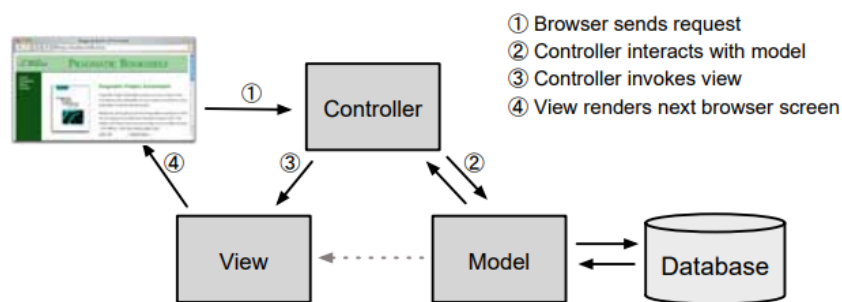


Figura 3.4. Descrição esquemática do padrão MVC.

Assim, o padrão de desenvolvimento na framework Django, representado na Figura 3.5., utiliza os seguintes componentes: *Model*, no qual são definidos todos os aspectos relacionados com os dados, ou seja, o acesso aos dados, a validação dos dados, o comportamento dos dados e as suas relações; *Template*, no qual são definidos todos os aspectos da apresentação das páginas aos usuários; e *Views*, no qual são definidos todos os aspectos das lógicas entre os *Models* e os *Templates*.

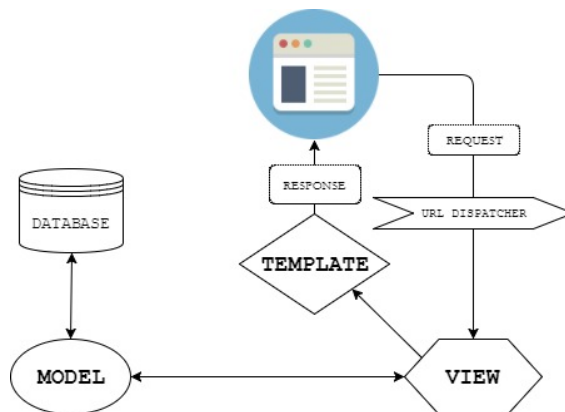


Figura 3.5. Descrição esquemática do padrão MTV.

3.3.2. Base de dados

Para comunicar com a base de dados *Django* usa uma *backend* para estabelecer a comunicação entre o *Model* e a base de dados. As *backend* que vêm pré-instaladas no *Django* são: *PostgreSQL*; *MySQL*; *SQLite3* e *Oracle*.

Tendo em conta o *rank* na Figura 3.6. e uma vez que o motor *Oracle* exige uma licença comercial, foi escolhida o *MySQL* como motor de base de dados para o portal *infantino*.

289 systems in ranking, December 2015

Rank			DBMS	Database Model	Score		
Dec 2015	Nov 2015	Dec 2014			Dec 2015	Nov 2015	Dec 2014
1.	1.	1.	Oracle	Relational DBMS	1497.55	+16.61	+37.76
2.	2.	2.	MySQL	Relational DBMS	1298.54	+11.70	+29.96
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1123.16	+0.83	-76.89
4.	4.	↑ 5.	MongoDB +	Document store	301.39	-3.22	+54.87
5.	5.	↓ 4.	PostgreSQL	Relational DBMS	280.09	-5.60	+26.09
6.	6.	6.	DB2	Relational DBMS	196.13	-6.40	-14.13
7.	7.	7.	Microsoft Access	Relational DBMS	140.21	-0.75	+0.31
8.	8.	↑ 9.	Cassandra +	Wide column store	130.84	-2.08	+36.78
9.	9.	↓ 8.	SQLite	Relational DBMS	100.85	-2.60	+6.15
10.	10.	10.	Redis +	Key-value store	100.54	-1.87	+12.66

Figura 3.6. Rank dos motores de base de dados em 2015 de <http://db-engines.com/en/ranking>

3.3.3. Servidor de páginas de internet

Para o servidor de páginas existem várias opções, o próprio *Django* vem com um servidor para testes simples, mas que não deve ser utilizado em produção. Uma vez que “Deploying Django with Apache and *mod_wsgi* is a tried and tested way to get Django into production *mod_wsgi* is an Apache module which can host any Python WSGI application,

including Django”(George, 2016, p. 316), escolheu-se o servidor de páginas de internet Apache2.

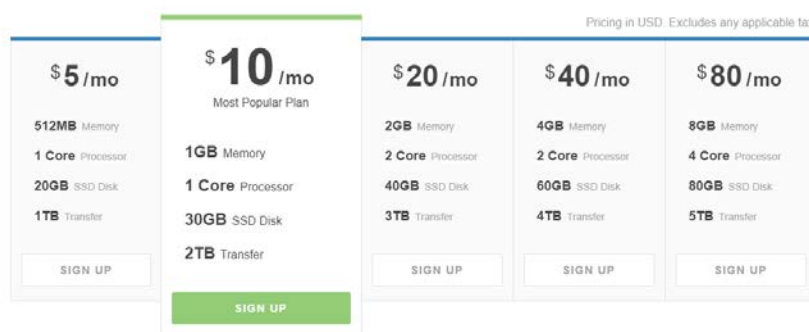
3.3.4. Sistema operativo do hospedeiro

Dada as escolhas feitas anteriormente e os requisitos não funcionais enunciados, o sistema operativo *Linux* parece ser o mais indicado. Diferentes distribuições deste sistema operativo podem ser escolhidas, mas aquelas que são mais acessíveis para o utilizador e que têm mais documentação disponível são Ubuntu e Debian sendo que, este último, foi escolhido.

3.3.5. Serviço de hospedagem

Entre os serviços de hospedagem possíveis foi escolhido o aluguer de um servidor de máquinas virtuais que permitisse: instalar diferentes distribuições *Linux*; tivesse um sistema de *backups*; e com pelo menos 1GB de RAM, uma vez que a base de dados da *MySQL* utiliza, com as predefinições de fábrica, cerca de 576MB. Foram comparados os serviços da *linode.com*, da *digitalocean.com* e da *dreamhost.com*: a *digitalocean.com* oferece serviços entre os 5 e os 80 dólares, Figura 3.7.; a *dreamhost.com* entre os 15 e os 120 dólares, Figura 3.8.; e a *linode.com* apresenta serviços entre os 10 e os 80 dólares, Figura 3.9..

Tendo em conta o custo, a *dreamhost.com* foi a primeira hipótese a ser descartada. O serviço de 5 dólares da *digitalocean.com* apenas oferece 512MB de memória RAM e, por isso, também foi descartado.



Pricing in USD. Excludes any applicable tax.

\$5/mo	\$10/mo Most Popular Plan	\$20/mo	\$40/mo	\$80/mo
512MB Memory	1GB Memory	2GB Memory	4GB Memory	8GB Memory
1 Core Processor	1 Core Processor	2 Core Processor	2 Core Processor	4 Core Processor
20GB SSD Disk	30GB SSD Disk	40GB SSD Disk	60GB SSD Disk	80GB SSD Disk
1TB Transfer	2TB Transfer	3TB Transfer	4TB Transfer	5TB Transfer
SIGN UP	SIGN UP	SIGN UP	SIGN UP	SIGN UP

Figura 3.7. Planos de máquinas virtuais oferecido pela *digitalocean.com* em dezembro de 2015

\$120/mo MAX OUT	\$60/mo 10x SAVER	\$30/mo MOST POPULAR	\$15/mo START VPS
8GB RAM	4GB RAM	2GB RAM	1GB RAM
240GB SSD Storage	120GB SSD Storage	60GB SSD Storage	30GB SSD Storage
Unlimited Bandwidth	Unlimited Bandwidth	Unlimited Bandwidth	Unlimited Bandwidth
Unlimited Domains	Unlimited Domains	Unlimited Domains	Unlimited Domains
Get This VPS	Get This VPS	Get This VPS	Get This VPS

Figura 3.8. Planos de máquinas virtuais oferecido pela dreamhost.com em dezembro de 2015

Linode 1GB	Linode 2GB	Linode 4GB	Linode 8GB
\$0.015/hr (\$10/mo)	\$0.03/hr (\$20/mo)	\$0.06/hr (\$40/mo)	\$0.12/hr (\$80/mo)
1 GB RAM	2 GB RAM	4 GB RAM	8 GB RAM
1 CPU Core	2 CPU Cores	4 CPU Cores	6 CPU Cores
24 GB SSD Storage	48 GB SSD Storage	96 GB SSD Storage	192 GB SSD Storage
2 TB Transfer	3 TB Transfer	4 TB Transfer	8 TB Transfer
40 Gbps Network In	40 Gbps Network In	40 Gbps Network In	40 Gbps Network In
125 Mbps Network Out	250 Mbps Network Out	500 Mbps Network Out	1000 Mbps Network Out
Sign Up	Sign Up	Sign Up	Sign Up

Figura 3.9. Planos de máquinas virtuais oferecido pela linode.com em dezembro de 2015

Assim, restava comparar o serviço de 10 dólares da *digitalocen.com* e o serviço de 10 dólares da *linode*. Depois de explorar as duas plataformas, escolheu-se utilizar o serviço da *linode.com*.

3.3.6. Alojamento dos ficheiros estáticos

Apesar da *framework Django* permitir que os ficheiros estáticos estejam no mesmo hospedeiro que os ficheiros dinâmicos, considera-se que este “*method is grossly inefficient and probably insecure, so it is unsuitable for production*”(“*Django documentation*,” 2015). Neste sentido, as opções apresentadas passavam pela configuração de outra VPS, para servir de servidor dedicado dos ficheiros estáticos, ou pela utilização de um serviço de armazenamento em cloud que “*lets you ignore the problems of serving static files and can often make for faster-loading webpages*”(“*Django documentation*,” 2015).

Uma vez que o Colégio Infante D. Henrique tem um serviço do *Google apps* e que seria desejável a integração do *login* no *infantinho.xyz* com o *e-mail* da *Google*, foi escolhido o serviço de alojamento de ficheiros na nuvem da Google.

3.4. Processo de desenvolvimento

Tendo em conta a natureza pedagógica deste projeto, os requisitos de desenvolvimento definidos, a sua dimensão, e uma vez que se pretende desenvolver uma “acção educativa [que] decorre de um estilo de trabalho sócio centrado nos alunos como actores dos seus percursos de aprendizagem realizados em cooperação”(Grave-Resendes & Soares, 2002, p. 42) foram atribuídos níveis de prioridade para o desenvolvimento das funcionalidades do portal:

- 1.º Blogue — importa criar circuitos de comunicação e divulgação dos trabalhos realizados pelos alunos, que, “no interior da turma e com o mundo exterior, promovem múltiplas interacções e dão sentido social imediato às aprendizagens.”(2002, p. 42).
- 2.º Registo de Presenças — este e outros registos fazem parte “das atividades de manutenção da sala de aula” que podem ser geridos pelos alunos uma vez que “a gestão cooperada pressupõe que todos saibam o que é preciso fazer e que cada um se responsabilize por uma tarefa.”(2002, p. 49)
- 3.º Listas de Verificação — estes registos individuais são confrontados com a avaliação do grupo e “a interacção que se estabelece entre a auto e a heteroavaliação ajuda à tomada de consciência daquilo que realmente cada um já sabe e daquilo que tem de trabalhar para melhorar”. Assim, neste processo comunicativo e cooperativo, “avalia-se não para classificar, discriminar, seleccionar, penalizar, mas para ajudar os alunos e o professor a direcionarem a aprendizagem e o ensino”.(2002, p. 141)
- 4.º Projetos — estes projetos decorrem dos interesses manifestados pelos alunos que, organizados em pequenos grupos, procuram uma explicação, ou mais informação, daquilo que gostariam de saber e ou produzir.
- 5.º Planos Individuais de Trabalho — são instrumentos de regulação e organização do trabalho de cada aluno, onde são registados os propósitos

de aprendizagem e de produção para um determinado período de tempo, e é com base neles que é feita uma avaliação do trabalho desenvolvido.

6.º Diários de Turma — funciona como um *feed* de uma rede social, no qual ficam registadas as interações sociais e as decisões da turma. É com base no Diário de Turma que, no Conselho de Cooperação, “geram-se consensos, tomam-se decisões, estabelecem-se e alteram-se regras de vida que a todos comprometem porque todos participaram na sua elaboração”(2002, p. 131).

7.º Recursos — os artefactos recolhidos ou produzidos pelos alunos e pelo professor tornam-se parte do acervo de materiais digitais disponíveis.

3.4.1. Configurações do Virtual Private Server

A interface de criação de servidores virtuais da *linode.com* tem alguns *scripts* de pré-instalação com as distribuições mais comuns de *Linux*. Foi escolhida a versão mais atual do *Debian*, no qual foi instalado apenas o *core* necessário para o sistema funcionar. De seguida procedeu-se à instalação dos pacotes necessários para o desenvolvimento com o *Django Framework* com os seguintes comandos com o utilizador *root*:

```
apt-get install python3 python3-pip build-essential libssl-dev libffi-dev python3-dev  
python3-venv apache2 libapache2-mod-wsgi-py3 mysql-server mysql-client  
libmysqlclient-dev git-core
```

O gestor de pacotes *aptitude* trata de instalar todas as dependências que estes pacotes possam ter.

3.4.1.1. Python Virtual Environment

Neste momento o servidor tem todos os pacotes necessários para a criação de um *python virtual environment* para o desenvolvimento do portal:

```
pip3 -m venv vpython
```

Passa-se a utilizar este *virtual environment* que contém todos os pacotes específicos do *Python* que serão utilizados para o desenvolvimento do portal *infantino* e que estão isolados do sistema operativo principal, permitindo que este ambiente possa ser replicado facilmente em qualquer máquina. Para ativar a utilização deste ambiente corremos:

```
source vpython/bin/activate
```

Assim, todos os comandos de instalação com o gestor de pacotes do *Python* instalarão estes pacotes apenas no ambiente virtual *vppython*.

Alguns pacotes foram instalados logo no princípio, pois já se sabia da sua necessidade nesta fase do desenvolvimento, e outros durante o processo, conforme as necessidades que surgiram. Inicialmente, instalou-se o *Django*, *django-storages*, *Google-cloud*, o *mysqlclient* e o *social-auth-app-django* com o seguintes comando:

```
pip install django django-storages Google-cloud mysqlclient social-auth-app-django
```

Com a instalação destes pacotes, o ambiente virtual está pronto para o desenvolvimento e para a implementação do portal *infantino.xyz*.

3.4.1.2. *Apache2 HTTP Server*

As páginas *html* geradas dinamicamente pelo *Django* são servidas aos utilizadores através de um servidor de *HTTP* que recebe os *HTTP requests* (*GET*, *HEAD*, *POST*, *PUT*, ...) e encaminha-os para o *Django*, que depois de os interpretar, devolve uma pagina *web* dinamicamente criada e que é lida pelo *Apache2* e enviada através de *HTTP responses*.

De modo a *informar* o *Apache2* da existência do *Django* e de que tipo de resposta deve dar, foi preciso editar o ficheiro */etc/apache2/sites-available/000-default.conf* com os dados da Tabela 3.3..

Tabela 3.3. Exemplo de Configuração do *Apache2*

1	<VirtualHost *:80>
2	ServerName infantino.xyz
3	ServerAlias www.infantino.xyz
4	WSGIDaemonProcess infantino python-home=/home/profigor/vpython python-path=/home/profigor/infantino
5	WSGIProcessGroup infantino
6	WSGIScriptAlias / /home/profigor/infantino/infantino/wsgi.py process-group=infantino
7	<Directory /home/profigor/infantino/infantino>
8	<Files wsgi.py>
9	Require all granted
10	</Files>
11	</Directory>
12	</VirtualHost>

3.4.1.3. *Base de Dados MySQL*

É necessário criar uma base dados para o projeto *infantino* e um utilizador com acesso local à base de dados criada:

```
mysql -u root -p
CREATE DATABASE database infdb;
```

```
CREATE USER 'infuser'@'localhost' IDENTIFIED BY 'infpassword';
GRANT ALL PRIVILEGES ON infdb . * TO 'infuser'@'localhost';
FLUSH PRIVILEGES;
Exit
```

O nome do utilizador criado para a base de dados, a palavra-passe e o nome da base de dados são as informações necessárias para a configuração do *Django*.

3.4.1.4. *Django Framework*

Esta *framework* de criação rápida de *webapps* permite o desenvolvimento ou a utilização de diferentes aplicações que se podem integrar num determinado *website*. Assim, o desenvolvimento de cada funcionalidade pode estar agregado a uma aplicação específica permitindo que o portal esteja funcional mesmo que nem todas as suas aplicações estejam completamente desenvolvidas.

O primeiro passo é criar o sítio agregador de todas as aplicações a serem instaladas, quer as desenvolvidas para o projeto, quer aquelas que serão utilizadas no projeto e que foram desenvolvidas por terceiros, i.e., *django-storages* e outras. Na Tabela 3.4. pode-se ver como a *framework* organiza a estrutura dos *websites* permitindo um rápido desenvolvimento e cuidando da maior parte do “*hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel*”(“*The Web framework for perfectionists with deadlines | Django,*” 2017).

Tabela 3.4. Lista dos Comandos Utilizados para a Criação da Estrutura do infantinho.xyz

comandos	estrutura criada
~/django-admin startproject infantinho	~/infantino/ manage.py infantino/ __init__.py settings.py urls.py wsgi.py
~/infantino/python3 manage.py startapp blog	~/infantino/blog/ __init__.py admin.py apps.py migrations/ __init__.py models.py

	tests.py views.py
~/infantino/python3 manage.py startapp record	~/infantino/record/ __init__.py admin.py apps.py migrations/ __init__.py models.py tests.py views.py
~/infantino/python3 manage.py startapp plan	~/infantino/plan/ __init__.py admin.py apps.py migrations/ __init__.py models.py tests.py views.py
~/infantino/python3 manage.py startapp heap	~/infantino/heap/ __init__.py admin.py apps.py migrations/ __init__.py models.py tests.py views.py
~/infantino/python3 manage.py startapp herd	~/infantino/herd/ __init__.py admin.py apps.py migrations/ __init__.py models.py tests.py views.py

Cada aplicação criada pelo Django já vem com uma estrutura que permite e orienta o trabalho para o modelo MTV, assim, o comando criou estes ficheiros:

- `manage.py` — um *script* de linha de comando que permite controlar várias opções e aceder a diferentes informações do sítio criado;

- `infantinho/__init__.py` — um ficheiro vazio que informa ao *Python* que esta diretoria deve ser considerada como um pacote *Python*;
- `infantinho/settings.py` — ficheiro com todas as configurações deste projeto;
- `infantinho/urls.py` — é o script responsável por tratar dos encaminhamentos para as aplicações instaladas (*URL dispatcher* na Figura 3.5);
- `infantinho/wsgi.py` — é o script que faz a comunicação entre o servidor de páginas da internet e o Django;

A framework cria também uma pasta para cada aplicação, nessa pasta estão todos os ficheiros necessários para o seu funcionamento organizados com o processo MTV.

O ficheiro `models.py` contém as classes dos modelos da aplicação, como na classe *Person* exemplo da Tabela 3.5. Cada classe representa uma tabela e cada campo representa uma coluna que são criadas automaticamente na base de dados, assim, no caso do exemplo apresentado, seria criada a tabela `app_person`, com três colunas *id*, que tem uma relação de um para um com as entradas na tabela *User*, padrão do Django; *first_name* e *last_name*,

Tabela 3.5. Exemplo de uma classe do Ficheiro `models.py`

```
4 class Person(models.Model):  
5     id = models.OneToOneField(User)  
6     first_name = models.CharField(max_length=30)  
7     last_name = models.CharField(max_length=30)
```

O ficheiro `views.py` é responsável pelas relações entre os modelos e os *templates*, i.e. pela lógica computacional por detrás da geração dinâmica das páginas *web*, na Tabela 3.6. podemos ver um exemplo de uma função que utiliza os dados da classe *Person* para verificar se a *id* existe na tabela de utilizadores do Django e retorna uma mensagem de erro, no caso de não o encontrar, ou uma mensagem de sucesso.

Tabela 3.6. Exemplo de uma função do Ficheiro `views.py`

```
8 @login_required(login_url='/login/Google-oauth2/?next=/')  
9 def hello_world(request):  
10     try:  
11         user = Person.objects.get(id=request.user.id)  
12     except ObjectDoesNotExist:  
13         messages.error(request, "Utilizador não está na Base de Dados")  
14     return render(request, 'hello.html')
```

```
15 username = '{} {}'.format(user.first_name, user.last_name)
16 messages.success(request, "Hello World")
17 return render(request, 'hello.html', username)
```

Os *templates* são por padrão armazenados numa pasta com o nome *templates/app* dentro da pasta da aplicação e são utilizados para gerar um ficheiro *HTML* de resposta. São criados com uma linguagem própria do *Django*, como se pode ver na Tabela 3.7., para se poderem servir de alguma lógica dinâmica e aceder aos dados passados pela *views.py*.

Tabela 3.7. Exemplo de um template

```
1 {% extends "index.html" %}
2 {% block content %}
3 {% if messages %}
4 <ul class="messages">
5 {% for message in messages %}
6 <li class="{{ message.tags }}">{{ message }}</li>
7 {% endfor %}
8 </ul>
9 {% endif %}
10 {% if username %}
11 <h1>Hello {{ username }}, Welcome Back!</h1>
12 {% endif %}
13 {% endblock content %}
```

Com a estrutura do projeto criada, foi necessário editar o ficheiro *infantino/settings.py* para configurar a base de dados, o hospedeiro dos conteúdos estáticos e o serviço de *login* integrado da *Google*.

3.4.1.4.1. Base de Dados

Para configurar a base de dados altera-se o ficheiro *settings.py* na sessão *DATABASES*, trocando o *backend* e completando com as informações de autenticação, como demonstrado na Tabela 3.8..

Tabela 3.8. Alterações ao Ficheiro *settings.py* para Configurar a Utilização do MySQL

```
78 DATABASES = {
79     'default': {
80         'ENGINE': 'django.db.backends.mysql',
81         'options': {
82             'init_command': "SET SQL_MODE='STRICT_TRANS_TABLES'",
83         },
84         'NAME': 'infdb',
85         'USER': 'infuser',
```

```
86     'PASSWORD': 'infpasword',
87     'HOST': 'localhost',
88     'CHARSET': 'utf8',
89     'COLLATION': 'utf8_general_ci',
90 }
91 }
```

3.4.1.4.2. Servidor de Ficheiros Estáticos

O *Django* vem pré-configurado para utilizar as pastas do sistema para servir os ficheiros estáticos, no entanto, a utilização de um servidor dedicado liberta grande parte do tráfego no sistema que serve os dados dinâmicos fazendo com que o portal tenha um comportamento mais otimizado. Neste sentido, o portal *infantino.xyz* utiliza o serviço da *Google Cloud Storage* com as configurações descritas na Tabela 3.8..

Tabela 3.9. Alterações ao Ficheiro settings.py para Ativar a Utilização do Google Cloud Storage

```
127 # Static files (CSS, JavaScript, Images)
128 STATICFILES_FINDERS = [
129     'django.contrib.staticfiles.finders.FileSystemFinder',
130     'django.contrib.staticfiles.finders.AppDirectoriesFinder',
131 ]
132 STATICFILES_DIRS = [
133     os.path.join(BASE_DIR, 'static'),
134 ]
135 credentials = service_account.Credentials.from_service_account_file(
136     os.path.abspath(os.path.join(BASE_DIR, 'key.json')))
137 DEFAULT_FILE_STORAGE = 'storages.backends.gcloud.GoogleCloudStorage'
138 STATICFILES_STORAGE = 'storages.backends.gcloud.GoogleCloudStorage'
139 GS_BUCKET_NAME = 'devinf'
140 GS_PROJECT_ID = '8asf4463803822356trg'
141 GS_CREDENTIALS = credentials
142 GS_AUTO_CREATE_BUCKET = 'True'
143 GS_AUTO_CREATE_ACL = 'publicRead'
144 GS_URL = '//storage.googleapis.com/%s' % GS_BUCKET_NAME
145 MEDIA_URL = GS_URL + '/media/'
146 MEDIA_ROOT = MEDIA_URL
```

```
147 STATIC_URL = GS_URL + '/static/'
```

3.4.1.4.3. Google OAuth2

O *Django* vem pré-definido para utilizar um *backend* próprio de autenticação sobre o qual pode ser criado um sistema de registo e de autenticação, mas, uma vez que todos os alunos do Colégio Infante D. Henrique têm uma conta *Google* gerida pela escola, torna-se preferível a utilização de um sistema integrado de *login* com os serviços da *Google*.

Neste sentido, foram criadas as chaves de autenticação, com o padrão *OAuth2*, na consola da *Google Cloud* e instalado um *backend* que faz a integração dos dois serviços. Estando instalado o '*social_django*', bastou proceder-se as alterações necessárias, descritas na Tabela 3.10., ao ficheiro *settings.py*.

Tabela 3.10. Alterações ao Ficheiro *settings.py* para Ativar a Utilização do Google OAuth2

```
154 # Authentication configs for social auth
155 AUTHENTICATION_BACKENDS = (
156     'social_core.backends.open_id.OpenIdAuth',
157     'social_core.backends.Google.GoogleOAuth2',
158     'django.contrib.auth.backends.ModelBackend',
159 )
160 LOGIN_REDIRECT_URL = '/'
161 SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = 'random.apps.googleusercontent.com'
162 SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = 'alçfj8483jalncvlakdçljf84589jalajghhs'
163 SOCIAL_AUTH_PIPELINE = (
164     'social.pipeline.social_auth.social_details',
165     'social.pipeline.social_auth.social_uid',
166     'social.pipeline.social_auth.auth_allowed',
167     'social.pipeline.social_auth.social_user',
168     'social.pipeline.user.get_username',
169     'social.pipeline.social_auth.associate_by_email',
170     'social.pipeline.user.create_user',
171     'social.pipeline.social_auth.associate_user',
172     'social.pipeline.social_auth.load_extra_data',
173     'social.pipeline.user.user_details',
174 )
175 SOCIAL_AUTH_DISCONNECT_PIPELINE = (
176     'social_core.pipeline.disconnect.allowed_to_disconnect',
177     'social_core.pipeline.disconnect.get_entries',
```

```
178 'social_core.pipeline.disconnect.revoke_tokens',  
179 'social_core.pipeline.disconnect.disconnect',  
180 )
```

3.4.1.5. Gestão de Versões

Para tornar o processo de desenvolvimento mais fluido e estabelecer um procedimento de trabalho que torne mais fácil testar o código, sem ter de desligar o servidor, criou-se um *clone* do ambiente de desenvolvimento numa máquina local. Assim, todo o desenvolvimento e teste é realizado nessa máquina e, quando o código está estável, é transferido para o servidor remoto.

A gestão das versões de código e da implementação de código é feita através da ferramenta *git*, no beanstalkapp.com, que permite a definição de *scripts* de implementação automático, com base em *triggers* definidos. No caso do projeto *infantino*, sempre que uma versão é carregada para a *branch master*, o código é atualizado no servidor remoto. Por outro lado, as versões de desenvolvimento vão sendo carregadas para a *branch devel*.

No servidor remoto executa-se:

```
git config --global user.name "Igor Almeida"  
git config --global user.email "prof.igor@colégioinfante.info"  
cd ~/infantino  
git init  
git remote add origin git@infantino.git.beanstalkapp.com:/infantino/infantino.git  
git fetch  
git checkout -b master --track origin/master
```

Por sua vez, no servidor local, depois de criada no portal do *beanstalkapp* uma *branch* com o nome *devel*:

```
git config --global user.name "Igor Almeida"  
git config --global user.email prof.igor@colégioinfante.info  
git clone -b devel --single-branch git@infantino.git.beanstalkapp.com:/infantino/infantino.git
```

No sítio do *beanstalkapp.com* configura-se um ambiente para trabalhar na pasta do servidor remoto e correr o seguinte código quando são carregados novos ficheiros para a *branch master*:

```
git pull origin master
```

No servidor local pode-se fazer as alterações necessárias ao código e testá-lo conforme for preciso. Sempre que se quiser guardar as alterações na *branch devel* correm-se os seguintes comandos:

```
git add .  
git commit -m "Comentário às alterações que foram feitas para facilmente identifica-las"  
git push -u origin devel
```

Por sua vez, quando o código está pronto para implementação correm-se os seguintes comandos:

```
git add .  
git commit -m "Comentário às alterações que foram feitas para facilmente identifica-las"  
git push origin devel:master  
git push -u origin devel
```

3.4.2. Blogue

Um dos principais objetivos do portal é servir como plataforma de divulgação dos trabalhos realizados pelos alunos, dando assim um maior sentido aos trabalhos dos alunos e valorizando o seu papel ativo na escola. Em termos do *design* do portal parece apropriado que o blogue seja a página de entrada do *infantinho.xyz*, pelo que, os visitantes do portal acedem diretamente aos trabalhos divulgados.

Para o desenvolvimento do blogue seguiu-se o tutorial *Try Django 1.9* (CodingEntrepreneurs, 2016) que serviu, por um lado, como um primeiro contacto com o *Django* e, por outro, como uma base para o blogue do *infantinho*. Desta forma, considerou-se mais importante o conhecimento adquirido ao longo processo de construção da versão inicial do blogue do que a rapidez da sua implementação.

Criou-se um *template* para a página inicial do blogue, Figura 3.10., que utiliza a framework de *HTML*, *CSS* e *JavaScript Bootstrap* de forma a que este *template* fosse sensível ao tamanho do dispositivo que o estiver a aceder. Deste modo, é possível uma apresentação apropriada ao tamanho da janela independentemente do dispositivo utilizado. Escolheram-se cores e ícones que fossem apelativos aos alunos do pré-escolar e primeiro ciclo e procurou-se providenciar hiperligações para alguns programas utilizados nas aulas e para algumas páginas da internet consultadas frequentemente.

Quanto às *views*, criaram-se as necessárias para preencher as funções previamente definidas nos requisitos, ou seja, *views* para visualizar, excluir, inserir e alterar mensagens no blogue. Cada *view* aponta para um *template* diferente. Para a função de visualizar criaram-se 3 versões diferentes, uma que funciona com as cartas do *Bootstrap*, em *grid*, outra em lista e, uma última, detalhada.

Implementou-se também um *JavaScript* do *Facebook*, Tabela 3.11., para habilitar um botão de *gosto* e um outro para *partilhar*.

Tabela 3.11. JavaScript para gosto e partilhar

```
1 <script>
2 window.fbAsyncInit = function() {
3   FB.init({
4     apId      : 'your-app-id',
5     autoLogAppEvents : true,
6     xfbml     : true,
7     version   : 'v2.10'
8   });
9   FB.AppEvents.logPageView();
10 };
11
12 (function(d, s, id){
13   var js, fjs = d.getElementsByTagName(s)[0];
14   if (d.getElementById(id)) {return;}
15   js = d.createElement(s); js.id = id;
16   js.src = "//connect.facebook.net/pt_pt/sdk.js";
17   fjs.parentNode.insertBefore(js, fjs);
18 }(document, 'script', 'facebook-jssdk'));
19 </script>
```

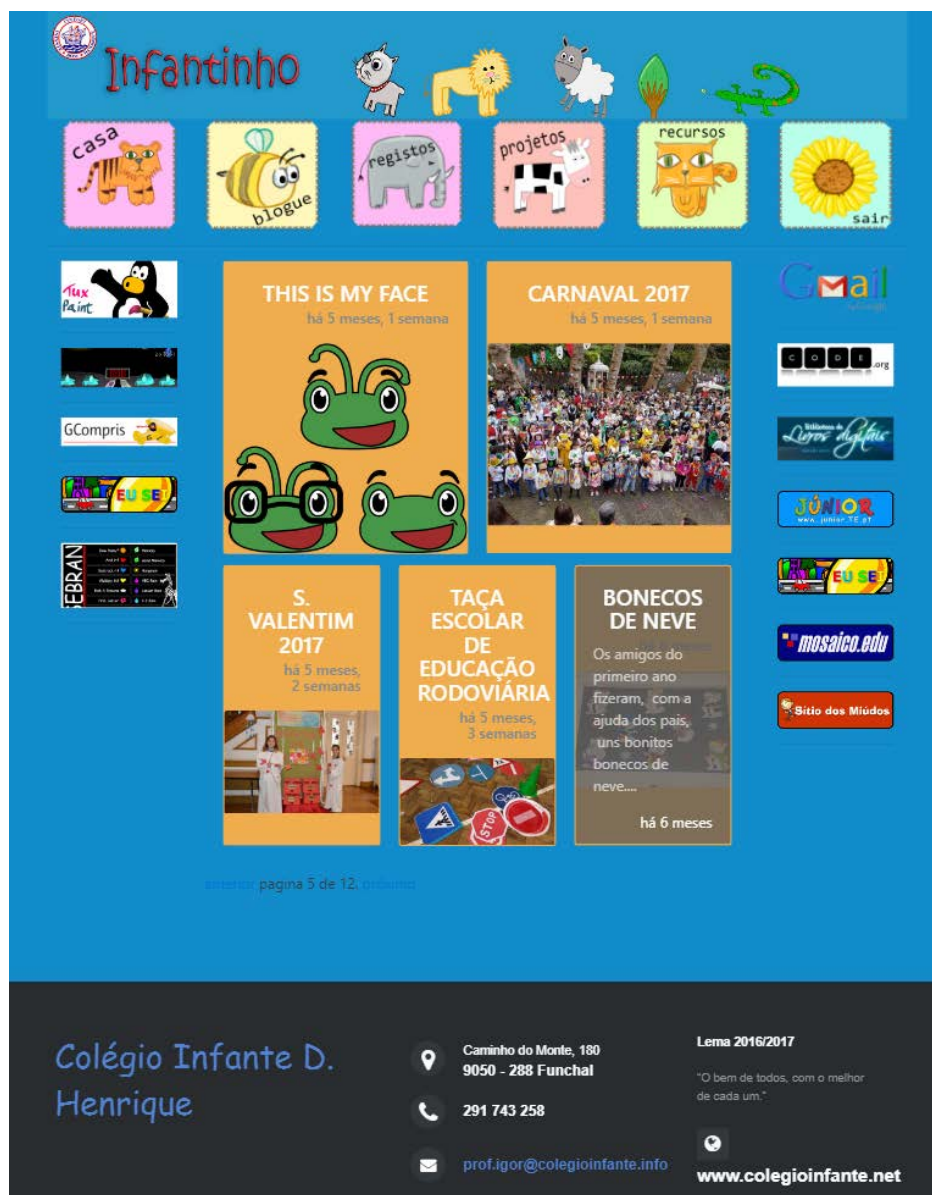


Figura 3.10. Página inicial do *infantino.xyz*

3.4.3. Registo de Presenças

A tarefa de marcar as presenças pode ser executada por qualquer aluno, desde o pré-escolar ao 4.º ano de escolaridade, desde que a interface criada para o efeito seja suficientemente intuitiva. No entanto, para se poder criar a *app* para marcar presenças é preciso que os alunos sejam utilizadores no portal e que existam na base de dados.

Neste sentido, decidiu-se criar uma *app* para a gestão de alunos, disciplinas e professores, visto o número de professores e disciplinas serem pequenos e, uma vez que o *django admin* permite a visualização, a exclusão, a inserção e a alteração dos dados das

tabelas das apps na base de dados, criou-se apenas uma *view* para carregar uma tabela com a informação dos alunos.

A tabela com a informação dos alunos tem a seguinte formatação: coluna 'A' com os números de alunos; coluna 'B' com os nomes dos alunos; e os nomes dos livros com as turmas. Deste modo, na própria *view*, Tabela 3.12., é feita a leitura da tabela e os alunos são inseridos na base de dados. O número de aluno é único e mantém-se durante toda a permanência do aluno no colégio, sendo por essa razão, que os e-mail dos alunos são criados com [número de aluno]@colegioinfante.info. A *view* da Tabela 3.12. associa cada aluno com o seu e-mail na base de dados dos utilizadores do portal e, quando os alunos tentam entrar o seu e-mail no portal, através do *OAuth2* da *Google*, o *Django* reconhece o e-mail e associa-o ao criado a partir da tabela.

Tabela 3.12. View para Carregar a Tabela e Criar um Utilizador para Cada Aluno

```
9 def upload_file(request):
10     if request.user.email != 'prof.igor@colegioinfante.info':
11         raise Http404
12     if request.method == 'POST':
13         form = UploadFileForm(request.POST, request.FILES)
14         if form.is_valid():
15             from openpyxl import load_workbook
16             f = request.FILES['file']
17             wb = load_workbook(filename=f, read_only=True)
18             school = {}
19             for book in wb:
20                 grade = book.title
21                 b = 1
22                 student = {}
23                 for row in book.rows:
24                     a = []
25                     for cell in row:
26                         a.append(cell.value)
27                     student[a[0]] = [b, a[1]]
28                     a = []
29                     b += 1
30                 school[grade] = student
31             for key in school:
32                 try:
33                     g = Grade(name=key.upper())
34                     g.save()
35                 except:
36                     pass
37             for v in school[key]:
38                 st_number = v
39                 st_number = str(st_number).rstrip('.')
40                 st_order = school[key][v][0]
```

```
41     st_name = school[key][v][1]
42     st_name = st_name.rstrip(' ')
43     f_name = st_name.split(' ', 1)[0]
44     namelist = st_name.split(' ', - 1)
45     l_name = namelist[len(namelist) - 1]
46     if User.objects.filter(username=st_number).exists():
47         continue # implementar user update para mudar de ano
48     else:
49         u = User.objects.create_user(f_name + '_' + st_number,
50                                     st_number + '@colegioinfante.info',
51                                     'aluno' + st_number,
52                                     )
53         u.first_name = f_name
54         u.last_name = l_name
55         u.save()
56         if Student.objects.filter(number=int(st_number)).exists():
57             continue
58         else:
59             u = User.objects.filter(username=f_name + '_' + st_number)
60             g = Grade.objects.filter(name=key.upper())
61             s = Student(
62                 name=u[0],
63                 number=st_number,
64                 list_number=st_order,
65                 grade=g[0],
66             )
67             s.save()
68         return HttpResponseRedirect('/add_students/')
69     else:
70         form = UploadFileForm()
71     return render(request, 'mng/upload.html', {'form': form})
```

Com os alunos e as turmas inseridos na base dados foram criados através da *app* de administração do *Django* as disciplinas, Figura 3.11., e o professor, Figura 3.12., e passou-se ao desenvolvimento da *app* para marcar as presenças.

Figura 3.11. Django Admin App – Criar Disciplina

Figura 3.12. Django Admin App – Criar Professor

Para a app de marcar presenças criou-se um único *Model*, Tabela 3.13., que comunica com a *app* de gestão dos alunos, como se pode ver no esquema das tabelas criadas na base de dados na Figura 3.13., para registar faltas dos alunos num determinado dia.

Tabela 3.13. Model da App que Guarda as Presenças na Base de Dados

```

5 class Presence(models.Model):
6     P = 0
7     F = 1
8     PCHOICES = ((P, 'P'),
9                  (F, 'F'),
10                  )

```

```

11 student = models.ForeignKey(Student)
12 is_absent = models.SmallIntegerField(default=0,
13                                     choices=PCHOICES)
14 date = models.DateField()
15 subject = models.ForeignKey(Subject)
16
17 def __str__(self):
18     return self.get_is_absent_display()

```

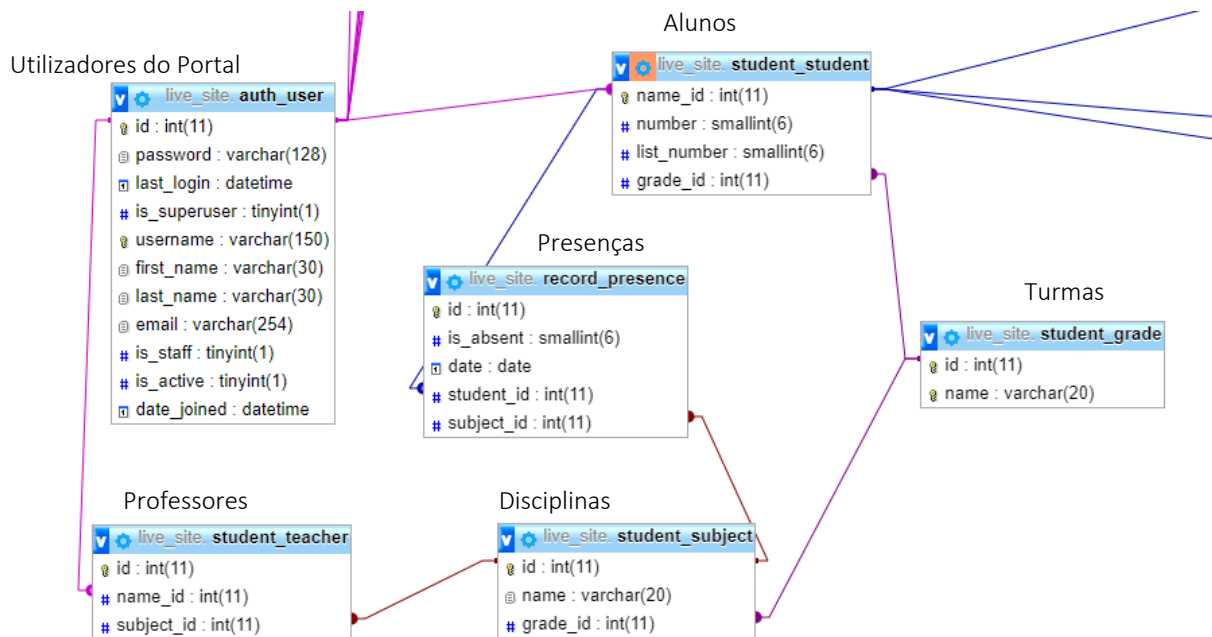


Figura 3.13. Tabelas Criadas pelo Django a Partir dos Models das Apps Student e Record

Criaram-se as views e templates para visualizar, inserir e transferir as presenças, com base no modelo de tabela .x/sx utilizada pelo Colégio Infante D. Henrique. No princípio da aula, o aluno responsável por marcar as presenças utiliza um dispositivo móvel e regista, numa página do *infantino.xyz*, Figura 3.14, os alunos que estão ausentes, clicando em P, para os presentes, ou em F, para os ausentes.

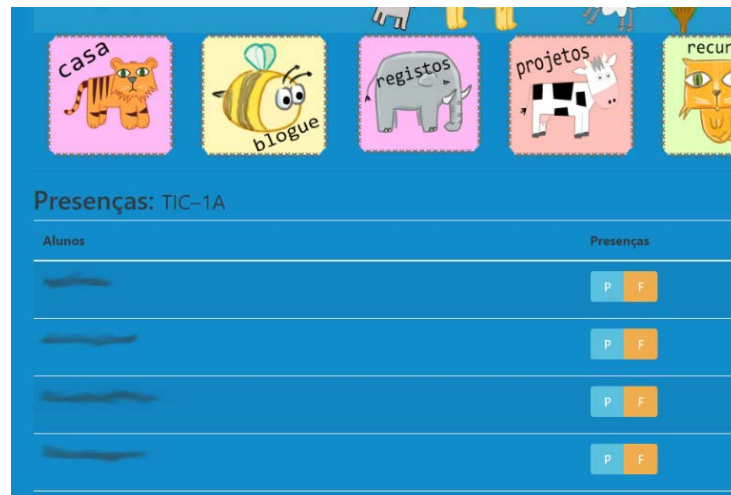


Figura 3.14. Marcar Presenças

3.4.4. Listas de Verificação

Para o desenvolvimento das listas de verificação começou-se por criar os *models*, Tabela 3.13., nos quais ficaram definidas as tabelas da base de dados para guardar a informação das autoavaliações dos alunos. Neste primeiro momento foi criada uma lista de verificação para os conteúdos de TIC, com os diferentes módulos definidos no DOM.

Tabela 3.14. Models das Listas de Verificação

```

5 class Modules(models.Model):
6     name = models.CharField(max_length=200)
7
8     def __str__(self):
9         return self.name
10
11
12 class Achiev(models.Model):
13     DIF = ((0, 'Nível 1'),
14           (1, 'Nível 2'),
15           (2, 'Nível 3'),
16           (3, 'Nível 4'),
17           (4, 'Nível 5'),
18           (5, 'Nível 6'),
19           (6, 'Nível 7'),
20           (7, 'Nível 8'),
21           (8, 'Nível 9'),
22           (9, 'Nível 10'),
23           )
24     module = models.ForeignKey(Modules)
25     name = models.CharField(max_length=200)
26     dif_level = models.SmallIntegerField(choices=DIF,
27                                         default=0)
28     student_achiev = models.ManyToManyField(Student,
29                                             through='StudentSysLevel')

```

```

30
31 def __str__(self):
32     return self.name
33
34 class Meta:
35     ordering = ['module']
36
37
38 class StudentSysLevel(models.Model):
39     AVAL = ((0, 'Ainda não consigo.'),
40            (1, 'Consigo com a ajuda do professor.'),
41            (2, 'Consigo com a ajuda de um colega.'),
42            (3, 'Consigo sozinho.'),
43            (4, 'Sou capaz de ajudar um amigo.'),
44            )
45     student = models.ForeignKey(Student)
46     achiev = models.ForeignKey(Achiev,
47                               verbose_name='')
48     student_level = models.SmallIntegerField(choices=AVAL,
49                                              default=0,
50                                              verbose_name='')
51     created_at = models.DateTimeField(auto_now=False,
52                                       auto_now_add=True)
53     updated_at = models.DateTimeField(auto_now=True,
54                                       auto_now_add=False)
55
56 def __str__(self):
57     return '{0} = {1}'.format(self.achiev, self.student_level)

```

Criaram-se *views* e *templates* para testar a utilização das listas, sendo que os alunos começam por escolher o módulo a ser avaliado, como na Figura 3.15..



Figura 3.15. Escolher o Módulo

De seguida, escolhem o nível que querem avaliar, Figura 3.16.. Os pontos atribuídos a cada nível são calculados pela *view*, adicionando-se 4 pontos para cada competência que pertence àquele nível de dificuldade.



Figura 3.16. Escolher o Nível de Dificuldade

Por fim, avaliam cada competência, Figura 3.17., numa escala de 0 a 4 sendo que:

- 0 pontos — *Ainda não consigo*, o aluno ainda não tentou ou não precisou daquele conhecimento;
- 1 ponto — *Consigo com a ajuda do professor*, o aluno já tentou, mas só conseguiu com a ajuda do professor;
- 2 pontos — *Consigo com a ajuda de um colega*, o aluno já conseguiu com a ajuda de um colega;
- 3 pontos — *Consigo sozinho*, o aluno já conseguiu concretizar a competência sozinho, sem ajuda; e
- 4 pontos — *Sou capaz de ajudar um amigo*, o aluno já foi capaz de ajudar um amigo.



Figura 3.17. Avaliar cada Competência Definida

3.4.5. Planos de Trabalho

Esta *app* é responsável pelos planos de projeto e pelos planos individuais de trabalho. Começou-se por definir quais os dados necessários para cada plano, criando-se os respetivos *Models*. Para o plano de projeto, criou-se uma *class* apenas com a informação

básica, i.e., título, descrição e grupo de trabalho, Tabela 3.15., a qual tem relações, *one-to-many*, com as outras classes criadas.

Tabela 3.15. Class Project

```
39 class Project(models.Model):
40     created_at = models.DateTimeField(auto_now=False,
41                                     auto_now_add=True)
42     updated_at = models.DateTimeField(auto_now=True,
43                                     auto_now_add=False)
44     title = models.CharField(max_length=50,
45                             blank=True,
46                             null=True)
47     description = models.CharField(max_length=200,
48                                  blank=True,
49                                  null=True)
50     group = models.ManyToManyField(Student,
51                                   blank=True)
52
53     def __str__(self):
54         return self.title
```

Criou-se a class **Knowwhat**, Tabela 3.16., com as informações: *o que querem saber, como vão encontrar a resposta às perguntas colocadas, quem faz, quando faz e a qual projeto pertence*. Deste modo, cada projeto pode ter várias entradas, *one-to-many*, na tabela do que querem saber, bem como nas outras classes criadas, ou seja, *o que já sabemos e apresentação*.

Tabela 3.16. Class Knowwhat

```
57 class Knowwhat(models.Model):
58     created_at = models.DateTimeField(auto_now=False,
59                                     auto_now_add=True)
60     updated_at = models.DateTimeField(auto_now=True,
61                                     auto_now_add=False)
62     what = models.CharField(max_length=200,
63                             blank=True,
64                             null=True)
65     how = models.CharField(max_length=200,
66                           blank=True,
67                           null=True)
68     when = models.DateTimeField()
69     project = models.ForeignKey(Project,
70                                on_delete=models.CASCADE)
71     who = models.ManyToManyField(Student,
72                                 blank=True,
73                                 null=True)
74
75     def __str__(self):
76         return self.wannaknow
```

As views e os templates para esta app dos planos estão em fase de desenho.

3.4.6. Diário de Turma e Recursos

Estas apps estão em fase de desenho. Por um lado, está a ser analisada a hipótese de se utilizar o classroom da Google integrado no portal infantinho, para o *diário de turma*. Por outro, e uma vez que as turmas já têm um diário de turma na sua sala de aula, um segundo diário, digital, na sala de TIC, poderá não ser necessário. Quanto à gestão dos recursos disponíveis e produzidos pelos alunos, estão a ser analisadas aplicações do *Django* que pudessem ser integradas no portal infantinho.xyz; uma hipótese é a utilização da *Google Cloud Storage* para armazenar os ficheiros e a criação de um sistema de categorização.

“Yoda: — No! Try not. Do, or do not. There is no try.”

(Kershner, Brackett, Kasdan, & Lucas, 1980)

4. Considerações Finais

O desenvolvimento deste portal foi um grande desafio, provavelmente o maior que alguma vez abracei. Os conhecimentos adquiridos durante o mestrado foram úteis, mas insuficientes para a abrangência deste projeto. Para o concretizar, novos conhecimentos eram necessários e não havia tempo a perder. Neste sentido, participei em alguns cursos online que foram uma grande ajuda para o bom desenrolar deste processo de desenvolvimento: *CS50: Introduction to Computer Science*, pela *Harvard University through edX*; *LFS101x.2: Introduction to Linux*, pela *Linux Foundation through edX*; e *Programming for Everybody (Getting Started with Python)*, *Python Data Structures*, *Using Python to Access Web Data*, *Using Databases with Python* e *Internet History, Technology, and Security*, pela *University of Michigan through Coursera*.

4.1. Conclusão

Na perspectiva do desenvolvimento ágil proposto, a maior parte dos objetivos foram concluídos. O portal está online e tem visitas frequentes de encarregados de educação, alunos e professores do Colégio Infante D. Henrique.

Analisando os objetivos propostos, importa referir que, no que diz respeito ao ***objetivo geral***, este não se completa com a versão do portal, criada até ao momento, pois estender-se-á pelo resto do processo de desenvolvimento que, tendo em conta o modelo de desenvolvimento profissional e pedagógico assumido, *Ação-Reflexão-Ação*, continuará.

Quanto aos ***objetivos específicos***, salienta-se que:

- ***Divulgação dos trabalhos*** — este objetivo foi concretizado na íntegra, fazendo parte do quotidiano dos nossos alunos. Neste momento, os alunos sabem que alguns trabalhos são divulgados no portal *infantino.xyz*, e nota-se um maior empenho por parte dos alunos.

- ***Organização e categorização dos artefactos pedagógicos produzidos*** — este objetivo ainda está a ser trabalhado e será implementado numa versão posterior do portal. No entanto, a sua funcionalidade foi testada com a utilização de pastas partilhadas através do *Google Drive*, o que serviu como uma história de utilização que poderá facilitar o desenvolvimento do código para esta funcionalidade para o portal.
- ***Criação, gestão e regulação de projetos de grupo*** — este objetivo ainda está a ser trabalhado e será implementado numa versão posterior do portal. No entanto, o desenvolvimento de cenários de aprendizagem foi implementado com recurso a tabelas que foram partilhadas com os alunos a partir do *Google Drive*, o que serviu como um teste dos campos necessários para a base de dados da *app* relacionada com este objetivo. Neste momento, estão a ser desenvolvidas as *views* e os *templates* para este objetivo.
- ***Criação, gestão e regulação dos planos de trabalho individual*** — este objetivo ainda está a ser trabalhado e será implementado numa versão posterior do portal. Os *Models* da *app* que será responsável pelos planos já foram definidos e estão a ser desenvolvidas as *views* e os *templates*.
- ***Regulação das presenças, aprendizagens e dos comportamentos*** — este objetivo já foi parcialmente implementado no portal atual, com a *app* para a regulação das presenças e com as listas de verificação. No entanto, mais trabalho é necessário para que se possa considerá-lo completo.

No que diz respeito às funcionalidades do portal que foram ordenadas de acordo com a importância que lhes foi atribuída, importa referir que as três primeiras (blogue, registo de presenças e listas de verificação) estão implementadas na versão atual do portal. A quarta e a quinta, planos de projetos e de trabalho individual, estão a ser desenvolvidas neste momento; os *Models* estão definidos e as *views* e *templates* estão em fase de desenho. Quanto às sexta e sétima funcionalidades, diário de turma e gestão dos recursos, estão em fase de reavaliação e a sua implementação será testada durante o próximo ano letivo.

4.2. Aplicabilidade

Para que este portal seja funcional, os alunos e o professor têm de abraçar a sua utilização diária. Neste sentido, é importante manter o desenho gráfico do portal atual e leve, pois a rapidez de resposta é fundamental para uma utilização agradável. A atualização dos serviços de hospedagem, por um lado, para máquinas virtuais com maior potência ou, por outro lado, para uma máquina local, poderá ser um caminho a seguir no sentido da otimização da resposta do servidor aos pedidos locais.

Os elementos pedagógicos da gestão cooperada da sala de aula funcionam como uma engrenagem e é difícil avaliar o seu impacto quando não estão todos inseridos num determinado contexto. Assim, é fundamental o desenvolvimento e a aplicação de todas as funcionalidades definidas para esta gestão, de modo a dar uma melhor resposta aos desafios que surgem no dia-a-dia.

Por fim, importa referir que o Colégio Infante D. Henrique, a partir do ano letivo 2017/2018, estará envolvido no Projeto de Autonomia e Flexibilização Curricular do Ministério da Educação. Ao mesmo tempo estão a ser desenhadas alterações de fundo no funcionamento das atividades de enriquecimento curricular, para as tornar menos estanques. Neste sentido, a utilização do portal infantinho.xyz poderá ser alargada a diferentes áreas do conhecimento, sendo que este portal poderá ser um bom recurso para toda a comunidade educativa.

Referências

- Bernabé, R. B., Navia, I. Á., & García-Peñalvo, F. J. (2015). Faat. *Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality - TEEM '15*, 687–694. <https://doi.org/10.1145/2808580.2808685>
- CodingEntrepreneurs. (2016). *Try Django 1.9*. Retrieved from https://www.youtube.com/watch?v=yfgsklK_yFo&list=PLEsfXfp6DpzQFqfCur9CJ4QnKQTVXUsRy
- Colégio Infante D. Henrique. (2015). Projeto Curricular de Escola. Retrieved April 25, 2017, from <http://www.colegioinfante.net/editor/userfiles/files/projectocurricularescola1415.pdf>
- Cranmer, S., & Perrotta, C. (2011). iTEC Scenario Development Process, (August). Retrieved from http://itec.eun.org/c/document_library/get_file?p_l_id=10307&folderId=36858&name=DLFE-1608.pdf
- Darling-hammond, L. (2006). CONSTRUCTING 21st-CENTURY TEACHER EDUCATION, 57(3). <https://doi.org/10.1177/0022487105285962>
- Diário da República, 1.ª série — N.º 65 — 4 de abril de 2016. (2016).
- Django documentation. (2015). Retrieved July 30, 2017, from <https://docs.djangoproject.com/en/1.9/>
- Django website. (2017). *The Web framework for perfectionists with deadlines*. Retrieved July 30, 2017, from <https://www.djangoproject.com/>
- Educativas, D. de S. de T. (2010). Documento de Orientações Metodológicas — Actividade de TIC na Educação Pré-Escolar e no Primeiro Ciclo do Ensino Básico. Funchal: Direcção Regional de Educação.
- Engeström, Y. (2016). *Studies in Expansive Learning: Learning What Is Not Yet There*. New York: Cambridge University Press.
- Freire, P., & Freire, A. na M. (2001). *Pedagogia dos sonhos possíveis*. São Paulo: Editora UNESP. Retrieved from <https://books.google.es/books?id=QxK-0mrXmtQC>
- George, N. (2016). *Mastering Django: Core*. Packt Publishing, Limited. Retrieved from <https://books.google.pt/books?id=71dHMQAACAAJ>

- Grave-Resendes, L., & Soares, J. (2002). *Diferenciação pedagógica*. Lisboa: Universidade Aberta.
- Hansson, D. H., & Fuchs, T. (2005). *Agile Web Development with Rails*. North. <https://doi.org/10.1109/WiCom.2008.2960>
- Jenkins, H. (2006). Confronting the Challenges of Participatory Culture: Media Education for the 21 Century. *Chicago, IL: The MacArthur Foundation*.
- Kershner, K. (Director), Brackett, L. (Screenplay), Kasdan, L. (Screenplay), & Lucas, G. (Story). (1980). *Star Wars: Episode V - The Empire Strikes Back*[Motion picture]. United States: A Lucasfilm Limited Production.
- Meyerovich, L. a., & Rabkin, A. S. (2013). Empirical analysis of programming language adoption. *Proceedings of the 2013 ACM SIGPLAN International Conference on Object Oriented Programming Systems Languages & Applications - OOPSLA '13*, (Section 7), 1–18. <https://doi.org/10.1145/2509136.2509515>
- Ministério da Educação. (2017). Perfil dos Alunos para o Século XXI. Retrieved from http://dge.mec.pt/sites/default/files/Noticias_Imagens/perfil_do_aluno.pdf
- Perrenoud, P. (2002). *A escola e a aprendizagem da democracia*. Porto: Edições ASA.
- Swain, N. R., Latu, K., Christensen, S. D., Jones, N. L., Nelson, E. J., Ames, D. P., & Williams, G. P. (2015). A review of open source software solutions for developing water resources web applications. *Environmental Modelling & Software*, 67, 108–117. <https://doi.org/10.1016/j.envsoft.2015.01.014>